НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

«На правах рукопису»

УДК _____

«До захисту допущено»
В.о. завідувача кафедри
_____ *О.А.Павлов*
(підпис) (ініціали, прізвище)
“___”_____2019 р.

Магістерська дисертація

зі спеціальності *121 «Інженерія програмного забезпечення»*

на тему: « Математичне та програмне забезпечення виявлення аномалій в потоках текстових даних»

Виконала Студентка VI курсу, групи *ІП-з81мп*

Афанасьєва Олена Євгенівна _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник *ст. вик. Олійник Юрій Олександрович* _____
(посада, науковий ступінь, , вчене звання, прізвище та ініціали) (підпис)

Консультант *доц., к.т.н., Ліщук К.І.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент *доц. каф. ТК, к.т.н., Остапенко К.Б.* _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

РЕФЕРАТ

Магістерська дисертація: 77 с., 21 рис, 33 таб., 2 додатки, 39 джерел.

Актуальність теми: Сьогодні важливість обробки текстових даних стрімко збільшується. Це пов'язано з великою кількістю текстової інформації, доступної через Інтернет. Оскільки мільйони символів вмісту формуються щодня, людина не має фізичної здатності обробляти всю інформацію.

На українському ринку поки відсутні застосунки для виявлення аномалій. Українські медіа, наукова сфера та бізнес все ще не мають інструменту для виявлення аномальних даних в текстах рідною мовою, що робить ці сфери менш розвинутими ніж такі ж сфери, що працюють у англомовному середовищі.

Мета дослідження: покращення аналізу україномовних потокових текстових даних та виявлення в них аномалій в режимі реального часу

Для реалізації поставленої мети були сформульовані **наступні завдання**:

- обґрунтувати вибір методу виявлення аномалій;
- створити математичну модель вибраного методу виявлення аномалій;
- виконати програмну реалізацію методу виявлення аномалій;
- дослідити ефективність методу виявлення аномалій.

Об’єкт дослідження: потоки україномовних текстових даних.

Предмет дослідження: виявлення аномалій в потокових текстових даних.

Методи дослідження: методи text mining, методи інтелектуального аналізу даних.

Наукова новизна: Найбільш суттєвими науковими результатами магістерської дисертації є:

- розробка адаптованого методу Isolation Forest виявлення аномалій в потоках текстових даних з підтримкою української мови.

Практичне значення отриманих результатів визначається тим, що запропонований модифікований алгоритм Isolation Forest, який підтримує виявлення аномалій в україномовних даних.

Зв'язок роботи з науковими програмами, планами, темами: робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Методи та технології високопродуктивних обчислень та обробки надвеликих масивів даних». Державний реєстраційний номер 0117U000924.

Апробація: Основні положення роботи доповідались і обговорювались на ІІІ всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019)

Публікації: Наукові положення дисертації опубліковані в Афанасьєва О.Є Виявлення аномалій в потоках текстових даних/ О.Є. Афанасьєва, Ю.О. Олійник // Матеріали ІІІ всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р.

Ключові слова: ПОТОКИ ДАНИХ, ВИЯВЛЕННЯ АНОМАЛІЙ, МЕТОД ІЗОЛЯЦІЙНОГО ЛІСУ, УКРАЇНОМОВНІ ДАНІ, ТЕКСТОВІ ДАНІ, ОБРОБКА ТЕКСТОВИХ ДАНИХ.

ABSTRACT

Master's dissertation consists 77 pages, 21 images, 33 tables, 39 referring sources.

Topicality: Today, the importance of textual data processing is increasing. This is due to the large amount of textual information available through the Internet. Because millions of content characters are generated every day, people do not have the physical ability to process all information.

The aim of the study: Improvement of real-time analysis of Ukrainian-language streaming text data and anomaly detection

To achieve this goal, the following tasks were formulated:

- justify the choice of anomaly detection method;
- to create a mathematical model of the chosen method of anomaly detection;
- to perform software implementation of the method of anomaly detection;
- to investigate the effectiveness of the anomaly detection method.

Object of study: streams of Ukrainian-language text data.

Subject of research: anomalies detection in streaming text data.

Research methods: text mining methods, data mining methods.

Scientific novelty: The most significant scientific results of the master's thesis are:

- development of an adapted Isolation Forest method for detecting anomalies in Ukrainian-language text data streams.

The practical value of the obtained results is determined by the fact that the proposed algorithm.

Relationship with working with scientific programs, plans, topics: work was performed at the Department of Automated Information Processing and Management Systems of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky» within the topic «Methods and technologies of high-performance computing and processing of large data sets». State Registration Number 0117U000924.

Testing: The main points of the work were reported and discussed at the Third All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies".

Publications: Scientific provisions of the dissertation published in Afanasieva O.E. Detection of anomalies in text data streams / O.E. Afanasieva, Y.O. Oliynyk // Proceedings of the Third All-Ukrainian Scientific and Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" (ISTU-2019) - Kyiv: NTUU “KPI them. Igor Sikorsky”, November 20-22, 2019.

Keywords: DATA FLOWS, ANOMALY DETECTION, ISOLATION FOREST METHOD, UKRAINOMATIC DATA, TEXT DATA, DATA MINING.

ЗМІСТ

ВСТУП 10
1 ОГЛЯД НАУКОВОЇ ЛІТЕРАТУРИ 12
1.1 Визначення потоку даних 12
1.2 Різниця між потоковими даними 13
1.3 Підтримка української мови 15
1.4 Передобробка даних 16
1.4.1 Очистка даних 17
1.4.2 Стемінг 17
1.4.3 Лематизація 19
1.4.4 Модель Bag of Words 20
1.4.5 Оцінка слів у словнику 21
1.5 Аномалії в потоках даних 22
1.6 Методи виявлення аномалій в потоках даних 23
1.6.1 Класифікація 29
1.6.2 Кластеризація 31
1.6.3 Статистичний аналіз 31
1.6.4 Алгоритм найближчого сусіда 32
1.6.5 Спектральні методи 32
1.6.6 Гібридні методи 33
1.7 Аналіз існуючих програмних засобів 33
1.7.1 Numeta 34
1.7.2 RapidMiner Starter Edition 34
1.8 Постановка завдання 35
Висновки до розділу 1 35
2 МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ 36
2.1 Модель потоку даних 36
2.2 Алгоритм ізоляційного лісу 37
2.2.1 Виявлення аномалій за допомогою ізоляційного лісу 37
2.2.2 Приклад роботи алгоритму 40
Висновки до розділу 2 44
3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 45
3.1 Опис стеку розробки 45
3.1.1 Мова програмування Python 45
3.1.2 Фреймворк Apache Spark 45

3.1.3 Бібліотеки 45
3.2 Моделювання програмного забезпечення 46
3.3 Архітектура програмного забезпечення 47
3.4 Керівництво користувача 49
Висновки до розділу 3 50
4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ ВИЯВЛЕННЯ АНОМАЛІЙ В ПОТОКАХ ТЕКСТОВИХ ДАНИХ 51
4.1 Опис експерименту 51
4.2 Результати експерименту 52
Висновки до розділу 4 56
5 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ 57
5.1 Опис ідеї 57
5.2 Технологічний аудит проекту 58
5.3 Аналіз ринкових можливостей запуску стартап проекту 59
5.4 Розроблення ринкової стратегії проекту 64
5.5 Розроблення маркетингової програми стартап проекту 66
Висновки до розділу 5 68
ВИСНОВКИ 69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 71
ДОДАТОК А СХЕМА СТРУКТУРНА АЛГОРИТМУ ISOLATION FOREST 76
ДОДАТОК Б СХЕМА СТРУКТУРНА КЛАСІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 77

ВСТУП

Кожного дня у світі публікується та генерується величезна кількість інформації. Людина вже не в змозі оброблювати таку кількість інформацію. Тому сьогодні однією з найактуальніших тем на ринку є програмні засоби, що можуть оброблювати великі масиви даних, що постійно оновлюється.

Виявлення аномалій в потоках даних в реальному часі сьогодні актуальна як ніколи. Виявлення гарячих трендів у медіа середовищі, пошук унікальних публікацій, очистка даних від похибок, інтернет-безпека – ці всі напрями використовують потребують автоматизованих програмних методів виявлення аномальних даних.

На світовому ринку існує багато сервісів, які надають змогу аналізувати дані на виявлення аномалій для бізнесу, медичної сфери, покращення результатів аналізу даних, аналізу сайтів новин та соціальних мереж для виявлення гарячих новин та трендів.

Про те всі ці сервіси не мають інструментів для обробки україномовних потокових текстових даних.

На українському ринку поки відсутні застосунки для виявлення аномалій. Українські медіа, наукова сфера та бізнес все ще не мають інструменту для виявлення аномальних даних в текстах рідною мовою, що робить ці сфери менш розвинутими ніж такі ж сфери, що працюють у англомовному середовищі.

Така ситуація склалася через відсутність у популярних засобах розробки для аналізу натуральних мов підтримки морфологічного аналізу для української мови.

Проте існують невеликі локальні інструменти для морфологічного аналізу даних українською мовою, які можна використати для розробки програмного забезпечення для виявлення аномалій в потоках текстових даних.

При поєднанні таких інструментів та одного з методів виявлення аномалій в потоках даних, стає можливим розробити математичне та програмне забезпечення для аналізу україномовних текстових даних.

Тому головною **метою дослідження** є *покращення аналізу україномовних потокових текстових даних та виявлення в них аномалій в режимі реального часу.*

До **об'єкту дослідження** відносяться *потоки текстових даних в режимі реального часу*, а до **предмету дослідження** відноситься *виявлення аномалій в потокових текстових даних*.

Науковою новизною роботи є *адаптований метод виявлення аномалій в потоках текстових даних з підтримкою української мови*.

1 ОГЛЯД НАУКОВОЇ ЛІТЕРАТУРИ

1.1 Визначення потоку даних

Потік даних (Data Stream у англо. літ.) – визначається як необмежена послідовність елементів даних або записів, які можуть як бути пов’язані між собою, так і не бути. Такі дані зазвичай позначаються часом та іноді географічною міткою [1]. Дані в потоці можуть передаватися з багатьох джерел.

Потокові дані також можна назвати потоком подій, оскільки кожен елемент таких даних є окремою подією у синхронізованій послідовності.

Потокові дані є великою проблемою для обробки класичними видами архітектури управління даними, які побудовані насамперед на концепції стійкості та статичних зборах даних. Через те, що найчастіше можливим є лише одноразове оброблення даних перед тим як отримати нову кількість даних, системи обробки потокових даних обробляють дані у режимі реального часу.

Неперервна обробка та управління даними – є типовою можливістю систем потокових даних.

Однак, великі розміри, різноманітність та швидкість великих даних створюють додаткові проблеми для цих систем, так як вони призначені для порівняно простих обчислень. Таких як: один запис за раз або набір об’єктів в короткий проміжок часу серед останніх даних.

Обчислення проводяться в реальному часі, іноді в пам’яті, та як самостійні обчислення.

Обробка компонентів зазвичай підключається до системи або джерела потоку неінтерактивним способом, тобто, не відправляючи назад жодних запитів та даних та не встановлюючи з ним взаємодії.

Концепція динамічного керування (concept of dynamic steering) передбачає динамічну зміну наступних кроків або напряму програми за допомогою безперервного обчислювального процесу з використанням потокової передачі. Динамічне керування часто є частиною потокового управління та обробки даних.

Проте всі додатки які працюють з потоками даних підпадають під цю категорію. Amazon Kinesis [2] та інші Apache [3] проекти з відкритим кодом як

Storm [4], Flink [5], Spark Streaming [6] та Samza [7] є прикладами систем для роботи з великими потоками даних.

Динамічне управління потоками даних у реальному часі та їх обробка є важливою частиною роботи в великими даними на сьогодні.

1.2 Різниця між потоковими даними

Після визначення, що таке потокові дані, визначимо чим відрізняються потокові дані та які існують проблеми в управлінні та обробці потокових даних.

Порівняємо “дані в русі” (in motion) та “дані в стані спокою” (at rest) [8], розрізнимо поточну та пакетну обробку даних та перелічимо їх проблеми.

Дані в стані спокою стосуються переважно статичних даних, зібраних з одного або декількох джерел даних, і аналіз відбувається після збору даних. Термін "дані в русі" відноситься до режиму, хоча застосовуються аналогічні методи збору даних, у якому дані аналізуються одночасно з їх генеруванням.

Так само, як обробка даних датчика в літаку або в самокерованому автомобілем. Аналіз адресованих даних називається пакетною або статичною обробкою, а аналіз потокових даних називається потоковою обробкою.

Час виконання та використання пам'яті більшості алгоритмів, які обробляють статичні дані, зазвичай залежать від розміру даних, і цей розмір легко обчислити з файлів або баз даних.

Ключовою властивістю потокової обробки даних є розмір необмежений розмір даних, і це змінює типи алгоритмів, які можна використовувати.

Алгоритми, які потребують циклічно повторення або циклу по всьому набору даних, неможливі, оскільки з потоковими даними неможливо дійти до кінця.

Моделювання та управління потоковими даними повинно включати обчислення на одному елементі даних або невеликому обсязі останніх елементів даних.

Ці обчислення можуть оновлювати показники, контролювати та візуалізувати статистику потокових даних. Або застосувати методи аналізу до потокових даних, щоб дізнатися динаміку зміни даних в залежності від часу.

Оскільки обчислення потрібно проводити в режимі реального часу, завдання аналізу, що обробляють потокові дані, повинні бути швидшими або не набагато довшими, ніж швидкість потоку даних. Що визначається за його швидкістю.

У більшості потокових систем система управління та обробки підписується на джерело даних, але не надсилає нічого назад до джерела потоку з точки зору зворотного зв'язку або взаємодії.

Вимоги до потокової обробки даних є зовсім іншими, ніж до пакетної обробки, де аналітичні кроки мають доступ до часто всіх даних і можуть зайняти більше часу для виконання складного аналітичного завдання з меншим тиском на час завершення окремих завдань управління та обробки даних.

Більшість організацій сьогодні використовують гібридну архітектуру. Іноді її називають як лямбда-архітектуру для обробки потокових та зворотних завдань одночасно. У цих системах потокова передача даних у режимі реального часу керується та зберігається до тих пір, поки ці елементи даних не будуть висунуті до пакетної системи та не стануть доступними для управління та обробки як пакетних даних.

У таких системах використовується прошарок для зберігання потоків, щоб увімкнути швидкі дерева потоків та забезпечити впорядкованість та послідовність даних. Обробний рівень даних використовується для отримання даних із рівня сховища для їх аналізу та, швидше за все, трохи пакетного потоку даних і повідомлення потокового сховища про те, що набір даних більше не потребує зберігання в потоковому сховищі. Головними проблемами даних, про які вказувалось, були масштабність, реплікація даних та довговічність, а також відмовостійкість у цьому типі даних є суттєвою.

Серед багатьох проблем є дві основні, які необхідно подолати, щоб уникнути втрати даних та включити аналітичні завдання в реальному часі. Одне завдання в потоковій обробці даних полягає в тому, що розмір і частота середніх даних можуть суттєво змінюватися з часом.

Ці зміни можуть бути непередбачуваними і можуть бути зумовлені поведінкою людини.

Наприклад, трансляція даних, знайдених у соціальних мережах, таких як Facebook [9] та Twitter [10], може збільшити обсяг під час свят, спортивних матчів або великих новинних подій.

Ці зміни можуть бути періодичними і виникати, наприклад, вечорами чи вихідними.

Наприклад, люди можуть розміщувати повідомлення у Facebook[9] більше ввечері, а не в денний робочий час. Потокові зміни даних також можуть бути непередбачуваними та спорадичними. Під час великих подій, спортивних матчів і подібних речей може збільшуватися розмір та частота даних. Інші зміни включають випадання або відсутність даних або навіть відсутні дані, коли виникають проблеми з мережею або пристрій, що генерує дані, має проблеми з обладнанням. Як приклад зміни потокової передачі даних можна розглянути кількість повідомлень у Twitter [10] в секунду. В середньому щодня надсилають 500 мільйонів повідомлень. Кожної секунди на цьому веб сервісі з'являється близько 5787 повідомлень в секунду, під час різних подій ця кількість може підвищуватися у кілька десятків разів [11].

Швидкість потоку повідомлень змінюється між різними часами та різними темами. Підводячи підсумок, потокові дані повинні оброблятися інакше, ніж статичні дані. На відміну від статичних даних, де можна визначити розмір, потокові дані постійно генеруються, і неможливо обробити їх всі одночасно.

Дані потоку можуть непередбачувано змінюватися як за розміром, так і за частотою. Зазвичай це пов'язано з поведінкою людини.

Нарешті, алгоритми для обробки потокових даних повинні бути відносно швидкими та простими. Оскільки невідомо, коли надійдуть наступні дані.

1.3 Підтримка української мови

pymorphy2 [12] – це морфологічний аналізатор та генератор для російської та української мови. Він використовує великі ефективно закодовані лексикони, побудовані з даних OpenCorpora [13] та LanguageTool [14]. Розроблено набір

лінгвістично мотивованих правил, що дозволяють проводити морфологічний аналіз та генерувати слова поза словниками, які спостерігаються в реальних документах.

Для російської pymorphy2 забезпечується найсучасніша якість морфологічного аналізу. Аналізатор реалізований мовою програмування Python з додатковими розширеннями C++. Акцент робиться на простоті використання, документації та розширюваності. Пакет поширюється за дозволеною ліцензією з відкритим кодом, заохочуючи його використання як в академічній, так і в комерційній обстановці.

Морфологічний аналіз – це аналіз внутрішньої структури слів. Для мов із багатою морфологією, як російська чи українська, використовуючи морфологічний аналіз, можна з'ясувати, чи може слово бути іменником чи дієсловом, чи може бути однини чи множини. Морфологічний аналіз є важливим етапом обробки природних мов.

Морфологічне покоління - це процес побудови слова з урахуванням його граматичного зображення; сюди входить лематизація, флексія та пошук лексеми слова.

pymorphy2 – це морфологічний аналізатор та генератор для української мови, що широко використовується в промисловості та в наукових колах. Він розробляється з 2012 року; Українська підтримка - це нещодавнє доповнення. Розробка свого попередника, pymorphy1, розпочалася в 2009 році. Пакет pymorphy2 за ліцензійною ліцензією (MIT), і він використовує дані відкритого дозволу, що мають дозвільну ліцензію.

1.4 Передобробка даних

Перед тим як виявляти аномалії в потоках текстових даних необхідно перед їх обробкою покращити набір даних за допомогою очистки даних, лематизації та стемінгу для отримання найбільш точних результатів.

1.4.1 Очистка даних

Проблеми з якістю даних виникають в окремих колекціях даних, як файли, бази даних, або потокові дані. Коли необхідно інтегрувати кілька джерел даних, потреба в їх очищенні значно зростає.

При потребі обробки неструктурованих текстових даних (таких як статті з сайтів новин, книжок, записів у соціальних мережах і т.д.), без попередньої очистки даних майже неможливо здійснити якісний аналіз даних.

Очистка даних використовується для покращення якості даних перед їх обробкою [15].

При неструктурованих текстових даних очистка даних потрібна для:

а) усунення нерелевантних символів (наприклад, будь-які символи окрім цифр та букв);

б) токенизації текстових даних на окремі слова;

в) видалення нерелевантних слів (таких як згадування в соціальних мережах та посилання на інші ресурси);

г) переведення усіх символів в нижній реєстр.

Після такої очистки даних можна приступати до нормалізації текстових даних за допомогою стемінгу або лематизації тексту.

1.4.2 Стемінг

Стемінг – це грубий евристичний спосіб пошуку основи слова. При такому підході основа слова не обов’язково має співпадати з морфологічний коренем заданого слова [16].

Існує кілька алгоритмів виконання стемінгу:

а) алгоритми пошуку;

Стемер шукає флективну форму у наданій таблиці пошуку.

До переваг такого підходу можна віднести: простоту, швидкість та легкість обробки виключень.

Серед недоліків даного алгоритму можна виділити: неможливість обробки нових та незнайомих слів, навіть якщо вони правильні, та завеликий розмір таблиці пошуку для ряду мов для мов зі складною морфологією.

Таблиці пошуку зазвичай генеруються у напівавтоматичному режимі, через що можуть генерувати неправильні форми для деяких слів (як неправильні дієслів у англійській мові).

Алгоритми пошуку можуть використовувати попередню автоматичну морфологічну розмітку, щоб уникнути однієї з помилок лематизації, при відношенні різних слів до однієї леми.

б) алгоритми відсічення закінчень;

Найпопулярнішим алгоритмом стемінгу є алгоритм відсічення закінчень. Він видаляє лишнє від кореня слова, що досить часто призводить до втрати словотворчих суфіксів.

Такі алгоритми не використовують довідникових таблиць, які складаються з флективних форм відношення кореня до та форми. Замість них зберігається список правил, який використовується алгоритмами, враховуючи форму слова, щоб відшукати його основу.

Рішення, які отримуються за допомогою алгоритмів відсічення, обмежуються тими частинами мови, які мають відомі закінчення та суфікси з деякими виключеннями. Це є великим обмеженням, так як не всі частини мови мають чітко сформульований набір правил. Лематизація намагається зняти це обмеження [17].

Також існують алгоритми відсічення префіксів, проте при використанні такого підходу, у мовних групах де присутні і префікси, і суфікси, використання такого підходу вимагає два рази оброблювати дані, що сповільнює швидкість роботи з ними.

Приклад, стемінгу для української мови можна побачити на Рисунку 1.1.

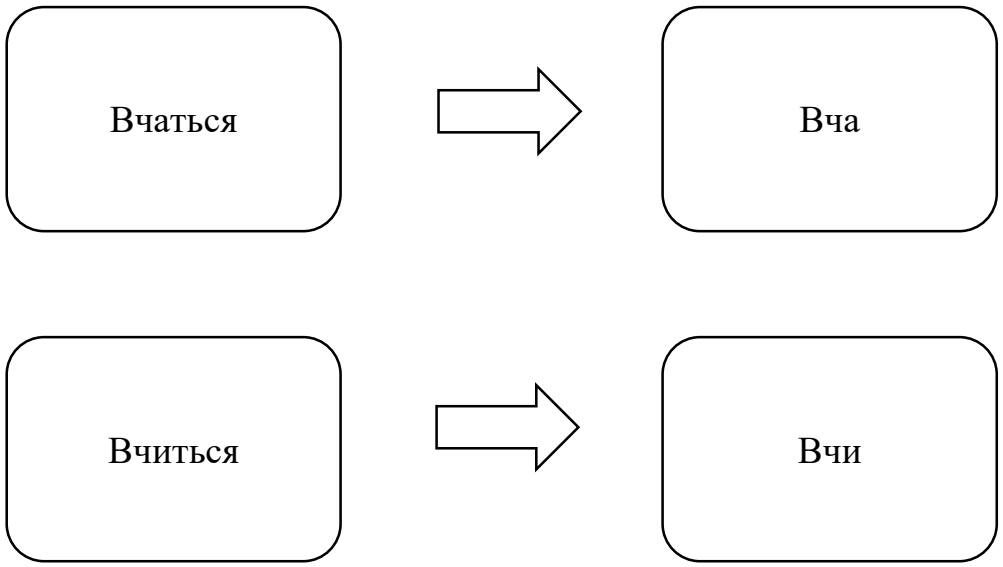


Рисунок 1.1 – Стемінг за алгоритмом відсічення закінчень

1.4.3 Лематизація

Лематизація – це більш тонкий спосіб нормалізації текстових наданих, який використовує словник та морфологічний аналіз, що знайти правильну форму – лему. Вона є більш складною в реалізації, ніж стемінг, проте дає більш точні результати.

Лематизація виконує зворотне перетворення форми слова, тобто заміняє граматичне закінчення суфіксом або закінченням початкової форми.

Також лематизація включає визначення частини мови слова та використання різних правил нормалізації для кожної частини мови. Визначення частини мови відбувається до спроби знайти основу слова, так як для деяких мов правила лематизації залежать від частини даного слова [12].

Цей підхід залежний від точного визначення лексичної категорії. Основна ідея закладається в тому, щоб при можливості отримати більше інформації про оброблювальному слові, то можна застосувати більш точні правила нормалізації.

Проте часткове спів падіння між правилами нормалізації для деяких лексичних категорій, вказання не вірної категорії або неможливість виділення правильної категорії нівелює переваги такого підходу щодо стемінгу.

Проте для слов’янських мовних груп лематизація працює більш точно ніж стемінг, що показано на Рисунку 1.1 та Рисунку 1.2.

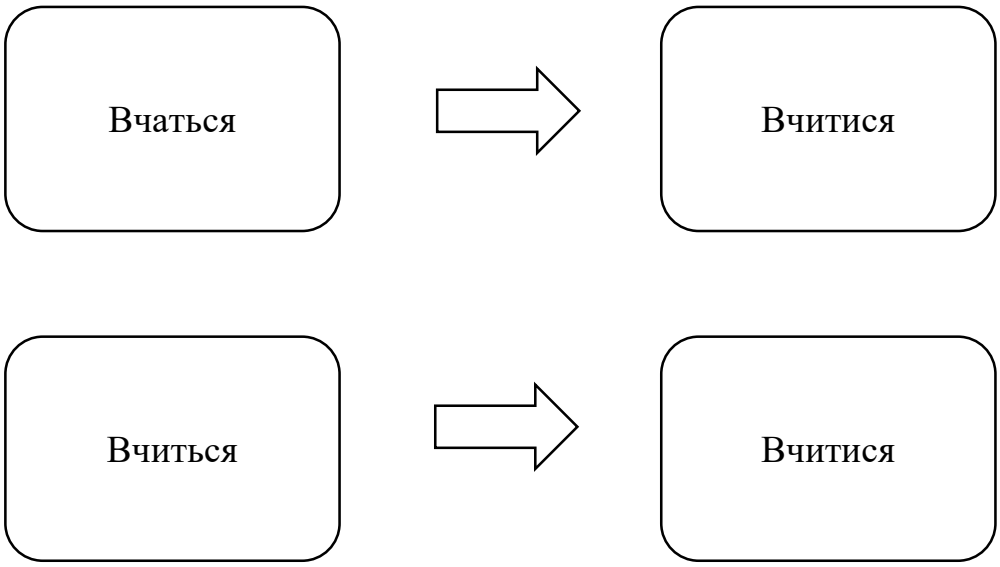


Рисунок 1.2 – Знаходження леми

1.4.4 Модель Bag of Words

Так як алгоритми машинного навчання не можуть працювати напряму з сирим текстом, необхідно конвертувати текст в набір цифр (так звані вектори). Даний процес називається витягом ознак.

Торба слів (Bag-of-Words Model) [18] – це проста техніка вилучення ознак, яка використовується при роботі з текстовими даними. Вона описує входження кожного слова в текст.

При використані даної моделі будь-яка інформація про послідовність або структуру слів ігнорується, саме через це назва даного підходу торба слів. Ця модель намагається зрозуміти, чи є задане слово у документі, але не знає де саме воно зустрічається.

Для використання такої моделі потрібно визначити словник відомих слів та обрати ступінь присутності відомих слів.

Складність цієї моделі в тому, як визначити словник та підрахувати входження слів.

Коли розмір словника збільшується, вектор документу збільшується також.

В деяких випадках можна отримати настільки великий об’єм даних, що вектор може складатися від тисячі до мільйонів елементів. При тому кожен оброблювальний документ може містити в собі лише невелику частину слів зі словника.

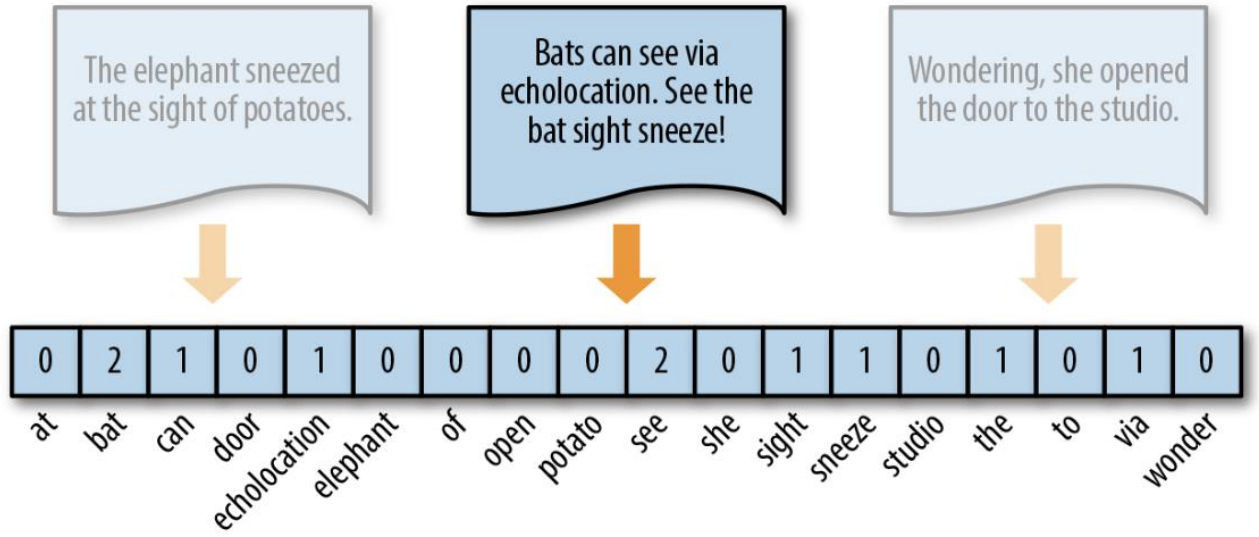


Рисунок 1.3 – Приклад роботи Bag of Words

Як наслідок, в векторному зображенні буде багато нулів, такі вектори називаються розрідженими векторами. Вони потребують більше пам’яті та обчислювальних ресурсів.

Саме для того, щоб зменшити кількість нулів в векторах здійснюється очистка та лематизація/стемінг текстових даних.

1.4.5 Оцінка слів у словнику

Після створення словника потрібно оцінити наявність слів. Серед методів оцінки можна виділити наступні три [19]:

а) бінарний підхід – приймається значення 1 якщо слово зустрічається та 0 якщо не зустрічається;

б) кількісний підхід – підраховується скільки разів слово зустрічається в документі;

в) частотний підхід – підраховується як часто кожне слово зустрічається в тексті відносно іншої кількості слів.

В частотній оцінці є проблема – слова з найбільшою частотністю мають найбільшу оцінку. В таких словах може бути не так багато інформаційного виграшу для моделі, ніж у словах, що зустрічаються рідше. Один зі способів виправити таку ситуацію – знижувати оцінку слова, яке часто зустрічається у всіх схожих документах, такий спосіб називається TF-IDF.

TF-IDF (term frequency – inverse document frency) [20] – це статична міра для оцінювання важливості слова в документі, який є частиною словника.

Оцінка по TF-IDF зростає пропорційно частоті появи слова в документі, але це компенсується кількістю документів, які мають це слово.

Типова формула оцінки слова X у документі У виглядає наступним чином:

$$w_{x,y} = tf_{x,y} * \log\left(\frac{N}{df_x}\right), \quad (1.1)$$

де $tf_{x,y}$ – частота x від y;

df_x – кількість документів, які містять x;

N – загальна кількість документів.

1.5 Аномалії в потоках даних

Аномалія – це відхилення поведінки системи від стандартної або очікуваної. Виявлення аномалій відноситься до пошуку непередбачених значень або патернів у потоках даних [21].

Аномалії в даних можуть відноситися до одного з трьох основних типів:

а) крапкові аномалії виникають в ситуації коли окремий екземпляр даних може розглядатися як аномальний у відношенні до інших даних;

б) відносні аномалії спостерігаються, якщо екземпляр даних є аномальним лише у заданому контексті, для виявлення такого типу аномалій основним є виділення контекстуальних (використовуються для визначення контексту кожного екземпляра) та поведінкових (визначають не контекстуальні характеристики, які відносяться до конкретного екземпляру даних) атрибутів;

в) колективні аномалії виникають, коли послідовність екземплярів даних є аномальною по відношенню до цілого набору даних, окремий екземпляр в такій послідовності не може бути аномальним.

В той час як крапкові та відносні аномалії можуть спостерігатися у будь-якому наборі даних, колективні спостерігаються лише у зв'язаних між собою даних.

Також крапкові та колективні аномалії можуть в той ж час бути і контекстуальними.

1.6 Методи виявлення аномалій в потоках даних

Існує кілька варіантів класифікації існуючих методів пошуку аномалій, серед яких найпоширеніші два види: по режиму розпізнання та за способом реалізації.

В залежності від застосовуваного алгоритму результатом роботи системи може бути мітка екземпляру як аномального або оцінка ступеню вірогідності того, що екземпляр є аномальним.

Процес виявлення аномалій може проводитися як на потоці даних, так і на архіві даних.

Часто для вирішення завдання пошуку аномалій потрібен набір даних, що описує систему. Кожен екземпляр в ньому описується міткою, яка вказує, чи є він нормальним або аномальним. Таким чином, безліч примірників з однаковими тегами формують відповідний клас.

Створення подібної промаркірованої вибірки зазвичай проводиться вручну і є трудомістким і дорогим процесом. В деяких випадках отримати екземпляри аномального класу неможливо в силу відсутності даних про можливі відхилення в системі, в інших можуть бути відсутні мітки обох класів. Залежно від того, які класи даних використовуються для реалізації алгоритму, методи пошуку аномалій можуть виконуватися в одному з трьох перерахованих нижче режимів:

а) розпізнавання з вчителем;

Дана методика вимагає наявності навчальної вибірки, повноцінно представляє систему і включає екземпляри даних нормального і аномального класів. Робота алгоритму відбувається в два етапи: навчання та розпізнавання. На першому етапі будується модель, з якої внаслідок порівнюються екземпляри, які не мають мітки. У більшості випадків передбачається, що дані не змінюють

свої статистичні характеристики, інакше виникає необхідність змінювати класифікатор.

Основною складністю алгоритмів, що працюють в режимі розпізнавання з учителем, є формування даних для навчання. Часто аномальний клас представлений значно меншою кількістю примірників, ніж нормальний, що може призводити до неточностей в отриманої моделі. У таких випадках застосовується штучна генерація аномалій.

б) розпізнавання частково з вчителем;

Вихідні дані при цьому підході представляють тільки нормальний клас. Навчившись на одному класі, система може визначати приналежність нових даних до нього, таким чином, визначаючи протилежний.

Алгоритми, що працюють в режимі розпізнавання частково з учителем, не вимагають інформації про аномальний клас, внаслідок чого вони ширше застосовуються й дозволяють розпізнавати відхилення за відсутності заздалегідь певної інформації про них

в) розпізнавання без вчителя.

Застосовується при відсутності апріорної інформації про дані. Алгоритми розпізнавання в режимі без вчителя базуються на припущенні про те, що аномальні екземпляри зустрічаються набагато рідше нормальних. Дані обробляються, найбільш віддалені визначаються як аномалії. Для застосування цієї методики має бути доступний весь набір даних, тобто вона не може застосовуватися в режимі реального часу [22].

До найвідоміших алгоритмів розпізнавання без вчителя можна віднести наступні:

а) Isolation Forest (ізоляційний ліс);

б) Angle-Based Outlier Detection (вугловий метод);

в) k-Nearest Neighbors Detector (k найближчих сусідів);

г) Histogram-based Outlier Detection (гістограмний метод);

д) Average KNN (метод k-середніх);

е) Cluster-based Local Outlier Factor (кластерний метод).

При порівняні цих методів використовувалася вибірка даних з продажів нерухомості [38], точковий графік розподілу даних якої зображено на Рисунку 1.4.

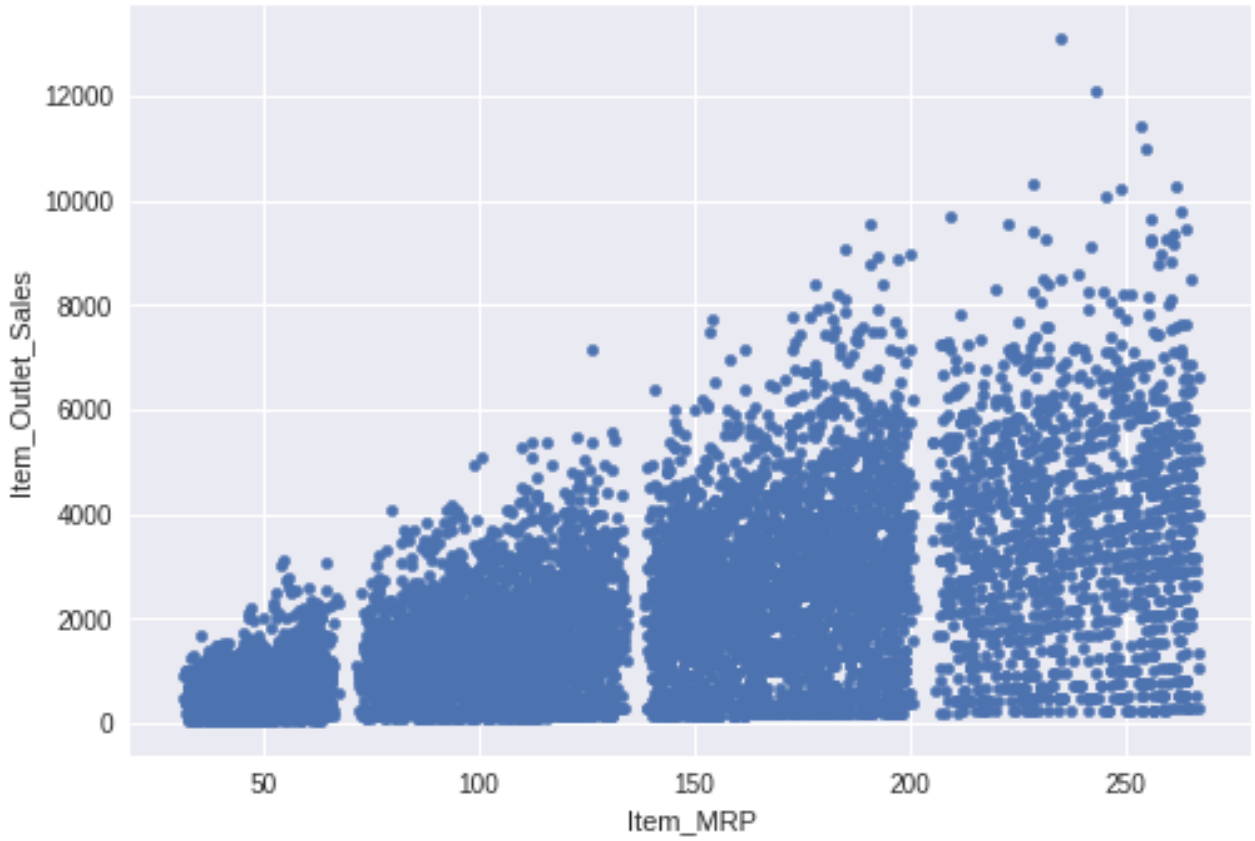


Рисунок 1.4 – Точкова діаграма розподілу тестових даних

Суть експерименту [37] полягає в тому щоб отримати точкову діаграму розподілу даних, на якій будуть зображені точки аномалій та отримати кількість аномальних та не аномальних даних.

Результати проведеного експерименту наведені нижче на Рисунках 1.5-1.10 та Таблиці 1.1.

а) Isolation Forest

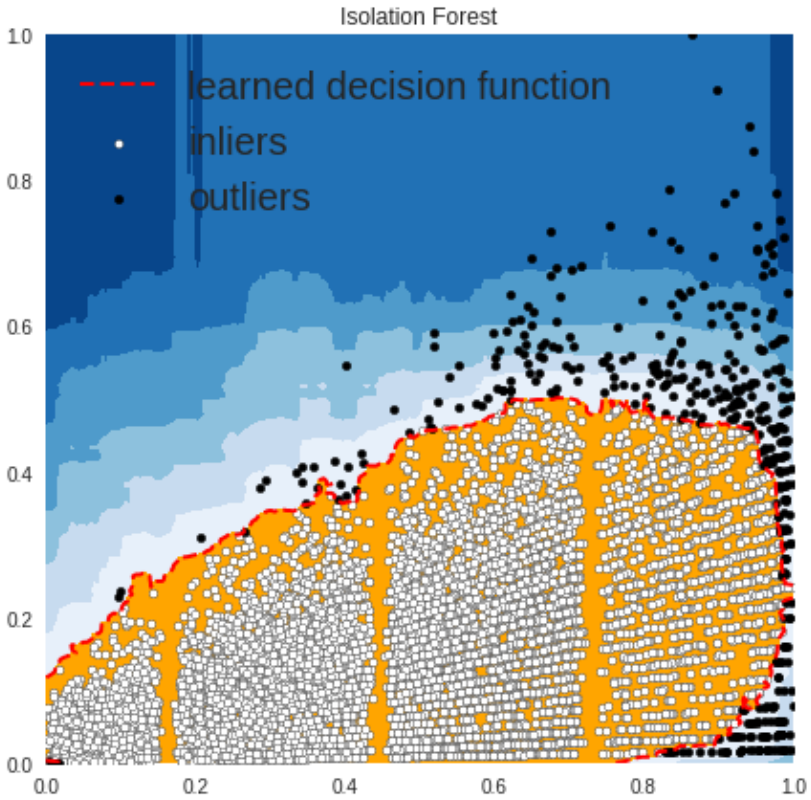


Рисунок 1.5 – Виявлення аномалій за алгоритмом Isolation Forest

б) Angle-Based Outlier Detection

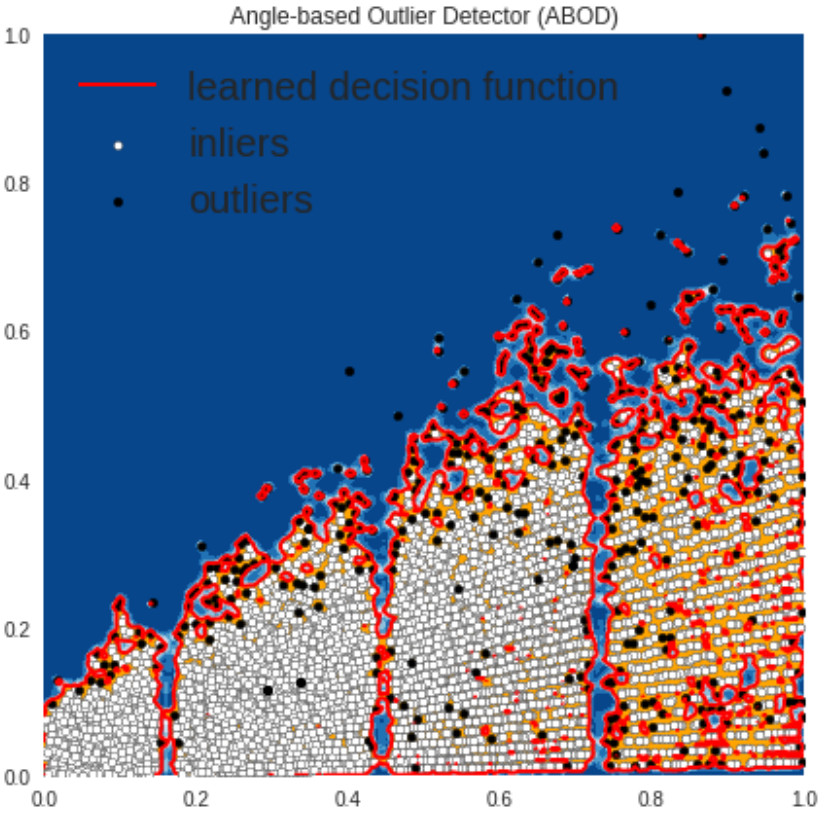


Рисунок 1.6 – Виявлення аномалій за алгоритмом Angle-Based

в) k-Nearest Neighbors Detector

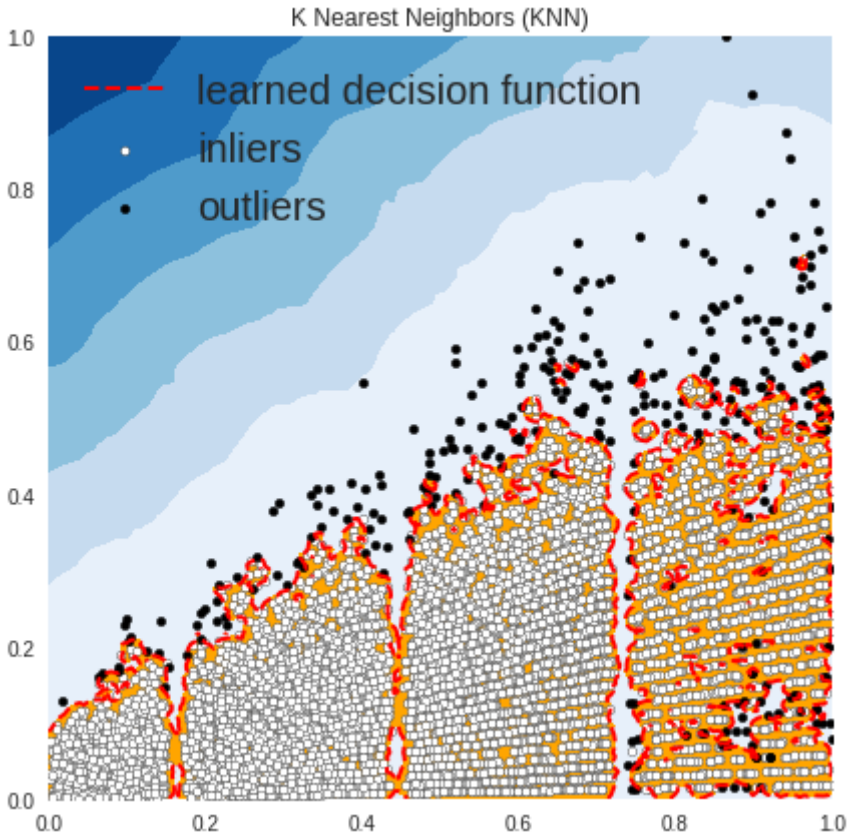


Рисунок 1.7 – Виявлення аномалій за алгоритмом найближчого сусіда

г) Histogram-based Outlier Detection

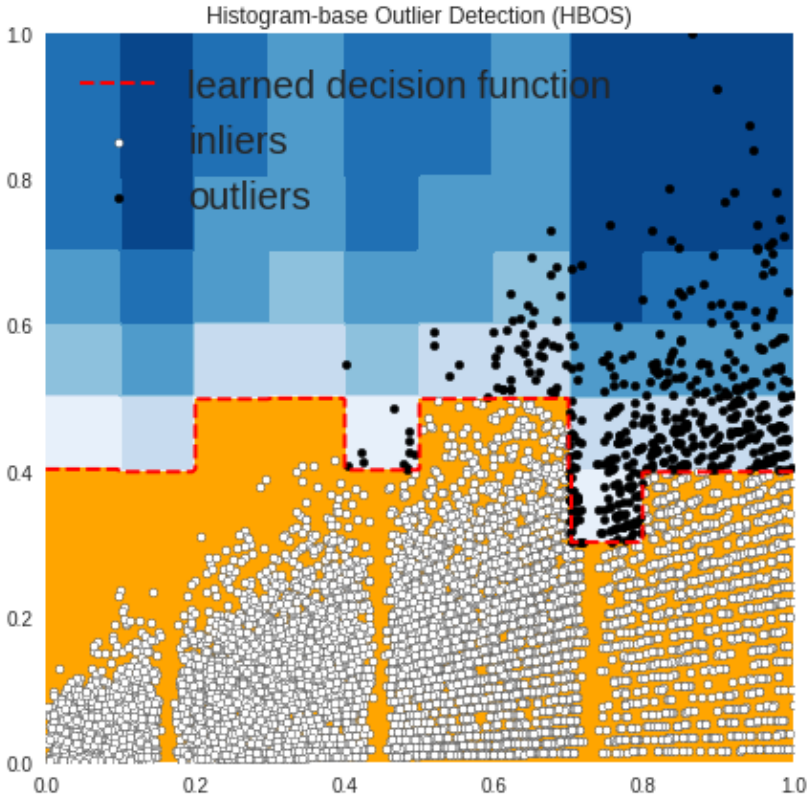


Рисунок 1.8 – Виявлення аномалій за алгоритмом гістограм

д) Average KNN

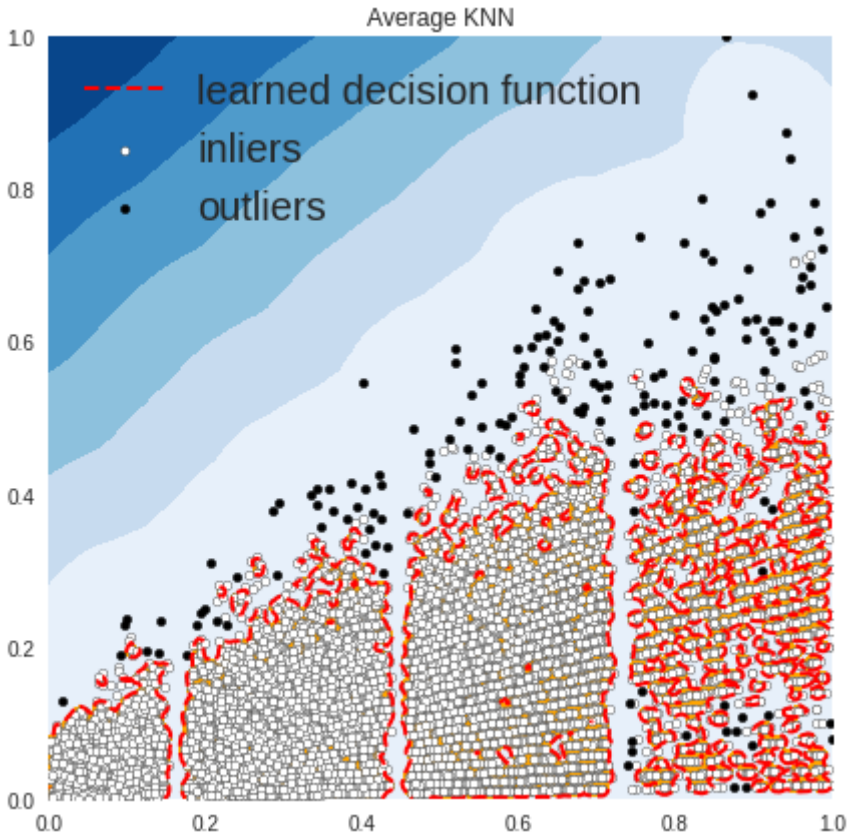


Рисунок 1.9 – Виявлення аномалій за алгоритмом K-середніх

е) Cluster-based Local Outlier Factor

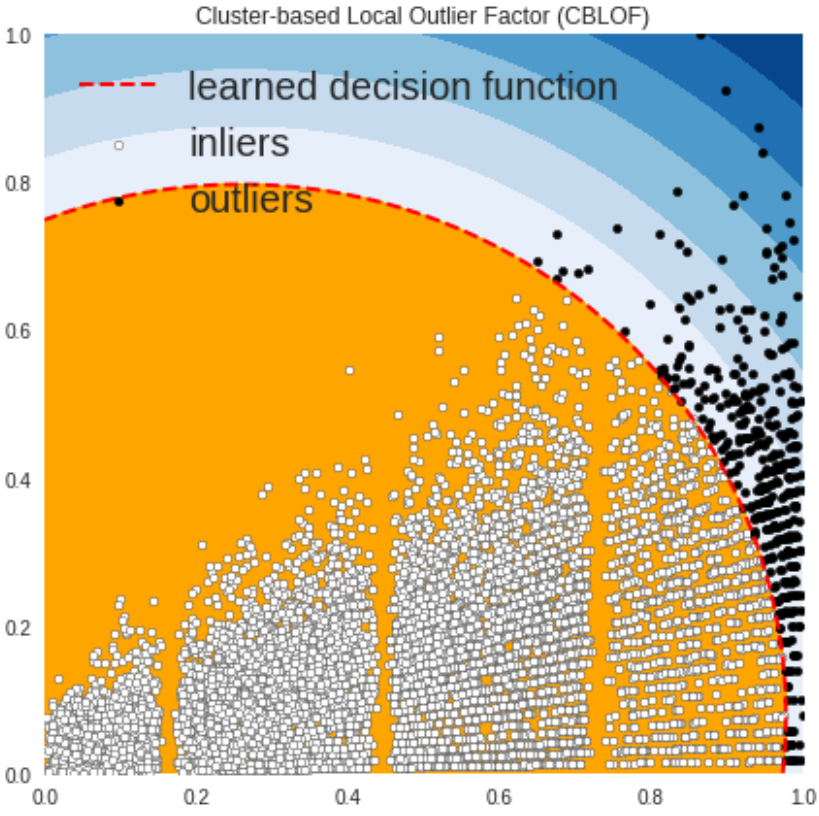


Рисунок 1.10 – Виявлення аномалій за алгоритмом кластеризації

Таблиця 1.1 – Результати експерименту

Назва методу	Аномальні дані	Звичайні дані
Isolation Forest	426	8097
Angle-Based Outlier Detection	447	8076
k-Nearest Neighbors Detector	310	8213
Histogram-based Outlier Detection	500	8023
Average KNN	175	8348
Cluster-based Local Outlier Factor	425	8098

1.6.1 Класифікація

Реалізація такого методу заснована на припущенні про те, що нормальна поведінка системи може визначатися одним або кількома класами. Таким чином, екземпляр даних, що не належить до жодного з класів, є відхиленням. Пошук аномалій проходить у два етапи: навчання та розпізнавання. Класифікатор навчається на масиві маркованих даних, далі визначається приналежність до одного з відомих класів. Якщо екземпляр даних не можна віднести до жодного з них, він позначається як аномалія.

Найбільш поширеними серед механізмів реалізації розпізнавання аномалій за допомогою класифікації є: нейронні мережі, Баєсові мережі, метод опорних векторів та метод на основі правил.

Метод виявлення аномалій на основі нейронних мереж включає два етапи. Перший: нейронна мережа навчається розпізнаванню класів нормальної поведінки на тренувальній вибірці. Другий: кожен екземпляр надходить в якості вхідного сигналу нейронної мережі. Система, заснована на нейронних мережах, може розпізнавати як один, так і кілька класів нормальної поведінки.

Для знаходження аномалій за допомогою лише одного класу використовуються реплікативні нейронні мережі. Технологія нейронних мереж глибинного навчання (Deep Learning), що одержала широке поширення, також успішно застосовується для вирішення даного завдання.

Байєсівської мережею є графічна модель, що відображає імовірнісні залежності безлічі змінних і дозволяє проводити імовірнісний висновок за

допомогою цих змінних. Вона складається з двох основних частин: графічна структура, яка визначає набір залежностей і незалежностей в безлічі випадкових величин, що представляють суб'єкти предметної області, і набір імовірнісних розподілів, що визначають силу відносин залежності, закодованих в графічній структурі. Таким чином, застосування Байєсової мережі при ідентифікації аномалій полягає в оцінці ймовірності спостереження одного з нормальних або аномальних класів.

Найбільш простою реалізацією цього підходу є наївний байєсівський класифікатор.

Метод опорних векторів (Support Vector Machine) застосовується для пошуку аномалій в системах, де нормальна поведінка видається тільки одним класом. Даний метод визначає межу регіону, в якому знаходяться екземпляри нормальних даних. Для кожного досліджуваного екземпляра визначається, чи знаходиться він в певному регіоні. Якщо екземпляр виявляється поза регіоном, він визначається як аномальний.

Останній метод ґрунтується на генерації правил, які відповідають нормальної поведінки системи. Примірник, який не відповідає цим правилам, розпізнається як аномальний. Алгоритм складається з двох кроків. Перший: навчання правил з вибірки за допомогою одного з алгоритмів, таких як RIPPER, Decision Trees і т.д. Кожному правилу привласнюється своє значення, яке пропорційно співвідношенню між числом навчальних примірників, що класифікуються, як правило, і загальним числом навчальних примірників, що покриваються цим правилом. Другий крок: пошук для кожного тестового екземпляра правила, яке найкращим чином підходить до даного екземпляру. Система може розпізнавати як один, так і кілька класів поведінки.

Одним з підвидів систем на основі правил є системи нечіткої логіки. Вони застосовується, коли межа між нормальним і аномальним поведінкою системи є розмитою. Кожен екземпляр є аномалією в деякій мірі віддаленості від центру мас нормального інтервалу [23].

1.6.2 Кластеризація

Дана методика припускає групування схожих екземплярів у кластери і не потребує даних про характеристики можливих відхилень. Виявлення аномалій може будуватися на наступних припущеннях:

а) нормальні екземпляри даних відносяться до кластеру даних, а аномалії не належать до жодного з кластерів, але при такому формулюванні може виникнути проблема визначення точних границь кластера;

б) нормальні дані ближчі до центру кластера, а аномальні - значно віддаленіші, але при такому формулюванні, коли аномальні екземпляри не є одиничними, вони можуть формувати нові кластери;

в) нормальні дані створюють великі та густі кластери, а аномальні - маленькі та різнозерні.

Однією з найлегших реалізації цього підходу на основі кластеризації - є метод K-середніх [24].

1.6.3 Статистичний аналіз

При використанні цього підходу досліджується процес, будується його модель, яка порівнюється з реальною поведінкою. Якщо різниця в реальній та ймовірною поведінкою системи, визначена заданою функцією аномальності, вище встановленого порогу, робиться висновок про наявність відхилень. Використовується припущення, що нормальна поведінка системи буде знаходитися у зоні високої ймовірності, в той час як аномалії - у зоні низької ймовірності [25].

Цей клас методів зручний ти, що не потребує попередніх знань про вид аномалії. Проте можливі складності при визначенні точного статистичного розподілення та порогу.

Методи статистичного аналізу підрозділяються на дві основні групи:

а) параметричні методи припускають, що нормальні дані генеруються параметричним розподіленням з параметрами θ та функцією щільності вірогідності $P(x, \theta)$, де x - спостереження. Аномалія є оберненою функцією

розподілу. Ці методи майже завжди базуються на Гаусовій або регресивній моделі, а також їх комбінації;

б) не параметричні методи припускають, що структура моделі не визначена, замість цього вона визначається з наданих даних. Включає методи на основі гістограм або функцій ядра.

Базовий алгоритм пошуку аномалій за допомогою гістограм включає два етапи. На першому відбувається побудова гістограми на основі різних значень обраної характеристики для екземплярів тренувальних даних. На другому етапі для кожного з досліджуваних екземплярів визначається приналежність до одного зі стовпчиків гістограми. Ті екземпляри, що не належать до жодної з гістограм помічаються як аномальні.

Розпізнавання аномалій за допомогою функції ядра відбувається аналогічно параметричним методам за виключенням способу оцінки щільності вірогідності.

1.6.4 Алгоритм найближчого сусіда

Для використання даної методики необхідно визначити поняття відстані між об'єктами. Прикладом може слугувати Евклідова відстань [26].

Два основних підходи базуються на наступних припущеннях:

а) відстань до k-го найближчого сусіда. Для реалізації цього підходу відстань до найближчого об'єкта визначається для кожного тестового екземпляра класу. Екземпляр аномалії найбільш віддалений від найближчого сусіда;

б) використання відносної щільності базується на оцінці щільності середовища кожного екземпляру даних. Екземпляр котрий знаходиться в середовищі з низькою щільністю оцінюється як аномальний, екземпляр оточений середовищем з високою щільністю оцінюється як нормальний. Для такого екземпляра даних відстань до його k-го найближчого сусіда еквівалентна радіусу гіпосфери з центром в цьому екземплярі і який містить k інших екземплярів.

1.6.5 Спектральні методи

Спектральні методи знаходять апроксимацію даних, використовуючи комбінацію атрибутів, які передають більшу частину варіативності в даних.

Цей метод заснований на наступному припущені: дані можуть бути вкладені у підмножину меншої розмірності, в якій нормальний стан і аномалії проявляються по іншому. Спектральні методи часто використовуються разом з іншими алгоритмами для обробки даних [27].

1.6.6 Гібридні методи

Гібридні методи розпізнавання аномалій дозволяють суміщати переваги різних підходів. При цьому різні техніки обробки можуть приміняться як послідовно, так і паралельно для досягнення усереднених результатів [28].

1.7 Аналіз існуючих програмних засобів

Основними аналогами розроблюваного математичного та програмного забезпечення є додатки, бібліотеки або фреймворки, які виявляють аномалії у потокових текстових даних українською в режимі реального часу. З цього виділяються наступні основні критерії підбору для схожого програмного забезпечення:

а) виявлення аномалій у потоковому наборі даних;

б) аналіз текстових даних;

в) аналіз даних українською мовою;

г) обробка даних в режимі реального часу.

Прямих аналогів розроблюваного додатку не виявлено, так як жоден зі знайдених додатків не оброблює дані, якщо вони надані українською мовою.

Розглянемо існуюче програмне забезпечення [29] з обробки потокових даних у Таблиці 1.1.

Серед наведеного вище програмного забезпечення найбільш наближеними до розроблювального додатку є Numenta та RapidMiner Starter Edition.

Таблиця 1.2 – Порівняння існуючих програмних засобів

Назва ПЗ	Обробка потоку даних у режимі реального часу	Виявлення та обробка аномалій	Аналіз текстових даних	Підтримка аналізу даних українською мовою
Numenta	+	+	+	-

Продовження таблиці 1.2

Avora	+	+	-	-
Splunk Enterprise	-	+	-	-
Loom Systems	+	+	-	-
Elastic X-Pack	+	+	-	-
Anodot	+	+	-	-
CrunchMetrics	+	+	-	-
Weka Data Mining	-	+	-	-
Shogun	-	+	-	-
RapidMiner Starter Edition	-	+	+	-
Dataiku DSS Community	-	+	-	-
ELKI	-	+	-	-
Scikit-learn	-	+	-	-

1.7.1 Numeta

Numeta [30] – це програмне забезпечення розроблене за технологією машинного навчання, що базується на теорії неокортексу. Ця технологія виявляє аномалії на серверах та додатках, поведінці людей, людській поведінці, гео-просторових трекінгових даних та для передбачення та класифікації природної мови.

Дане програмне забезпечення аналізує текстові дані на наявність аномалій в режимі реального часу, але не підтримує українську мову.

1.7.2 RapidMiner Starter Edition

RapidMiner Starter Edition [31] – це програмне забезбечення для бізнес-рішень для аналізу даних.

До плюсів даного програмного забезпечення відноситься аналіз текстових даних на виявлення аномалій, але додаток не аналізує дані в режимі реального часу та не підтримує аналіз для даних українською мовою.

В загальному на ринку відсутнє програмне забезпечення для аналізу великих потоків неструктурованих текстових даних українською мовою.

1.8 Постановка завдання

В рамках дослідження мають бути вирішені наступні завдання:

- обґрунтувати вибір методи виявлення аномалій;
- створити математичну модель вибраного методу виявлення аномалій;
- виконати програмну реалізацію методу виявлення аномалій;
- дослідити ефективність методу виявлення аномалій.

Висновки до розділу 1

Проблема виявлення аномалій в потоковому передачі даних має наступні характеристики:

- потік нескінченний, тому будь-які алгоритми поза лінійного навчання, які намагаються зберігати весь потік для аналізу, не вистачить місця в пам'яті;
- потік містить в основному звичайні випадки, оскільки аномальні дані рідкісні і можуть бути недоступними для тренувань. У цьому випадку будь-які класичні класифікатори, які потребують повністю маркованих даних, не підходять;
- потокові дані розвиваються з часом. Таким чином, модель повинна адаптуватися до різних частин потоку, щоб підтримувати високу точність виявлення;
- підтримка української мови у додатках, які виявляють аномалії у потоковій передачі даних не зустрічається.

В рамках цього розділу були сформовані та поставлені задачі, які має вирішувати розроблюване програмне забезпечення.

2 МАТЕМАТИЧНЕ ОБГРУНТУВАННЯ

2.1 Модель потоку даних

Потік даних $S = \{(m_0, t_o), (\dots), (m_i, t_i), (m_{i+1}, t_{i+1}), (\dots)\}$ – є нескінченним потоком даних, що надходять з одного або кількох джерел де отримана пара (m_i, t_i) означає, що повідомлення m_i отримане в час t_i [32].

У такому випадку часовим вікном W_i є інтервал часу фіксованого розміру δ, що починається у точці t_i.

Наведемо схему моделі потоку даних за часовим вікном на рис 2.1.

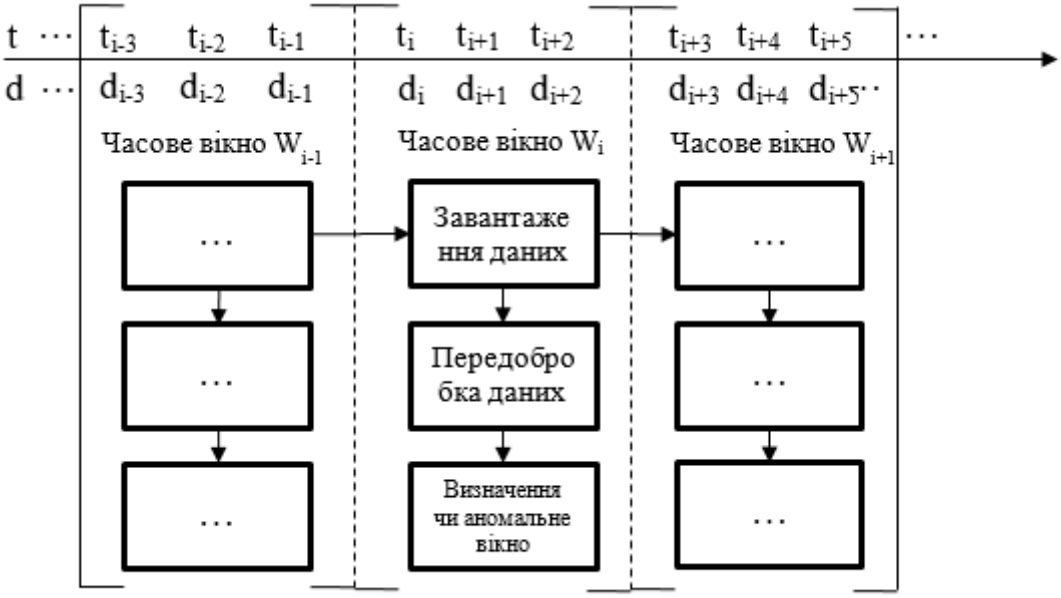


Рисунок 2.1 – Модель потоку даних

Наведемо алгоритм розбиття потоку даних на часові вікна:

Вхідні дані: S – потік даних, δ – фіксований інтервал часу

Вихідні дані: W_j – часове вікно

1. Встановлення границі часового вікна $g = t_i + \delta$
2. **for** $t_i \leq g$ **do**
3. $W_j \leftarrow W_j \cup S(d_i, t_i)$
4. **return** W_i

Після виділення часового вікна за допомогою алгоритму ізоляційного лісу визначаємо чи є вікно аномальним.

2.2 Алгоритм ізоляційного лісу

Метод ізоляційного лісу *(Isolation Forest або iForest)* [33] утворює скупчення ізоляційних дерев (Isolation Trees або iTrees) для заданого набору даних, в такому випадку аномальні дані – це випадки в яких середня довжина до iTrees коротка.

В такому методі є лише дві змінні: кількість дерев для будування та розмір вибірки.

iForest відрізняється від існуючих методів заснованих на моделі, відстані та щільності наступними способами:

а) ізоляція дозволяє будувати часткові моделі та використовувати під-вибірку так, як неможливо в існуючих методах, невеликий розмір вибірки дає кращі ізоляційні дерева оскільки ефекти блокування та маскування зменшуються;

б) ліс ізоляцій виключає відстань та щільність для виявлення аномалій, це виключає головні обчислювальні витрати на розрахунок відстані у всіх методах, що базуються на відстані та щільності;

в) ліс ізоляцій має лінійну часову складність з низькою константою та низькою потребою в пам'яті, найефективніший існуючий метод досягає лише приблизної лінійної складності в часі з високим рівнем використання пам'яті;

г) ліс ізоляцій має можливість для масштабування для обробки дуже великих об’ємів даних та аналізу даних з великою кількістю неактуальних атрибутів.

2.2.1 Виявлення аномалій за допомогою ізоляційного лісу

Виявлення аномалій за допомогою ізоляційного лісу зазвичай являється двоетаповим процесом. Перший (навчання) – побудова ізоляційних дерев використовуючи навчальний сет. Другий (тестування) етап проходить тестові екземпляри через ізоляційні дерева для отримання оцінки аномалії кожного екземпляра.

Приклад побудованого ізоляційного лісу наведено на Рисунку 2.2.

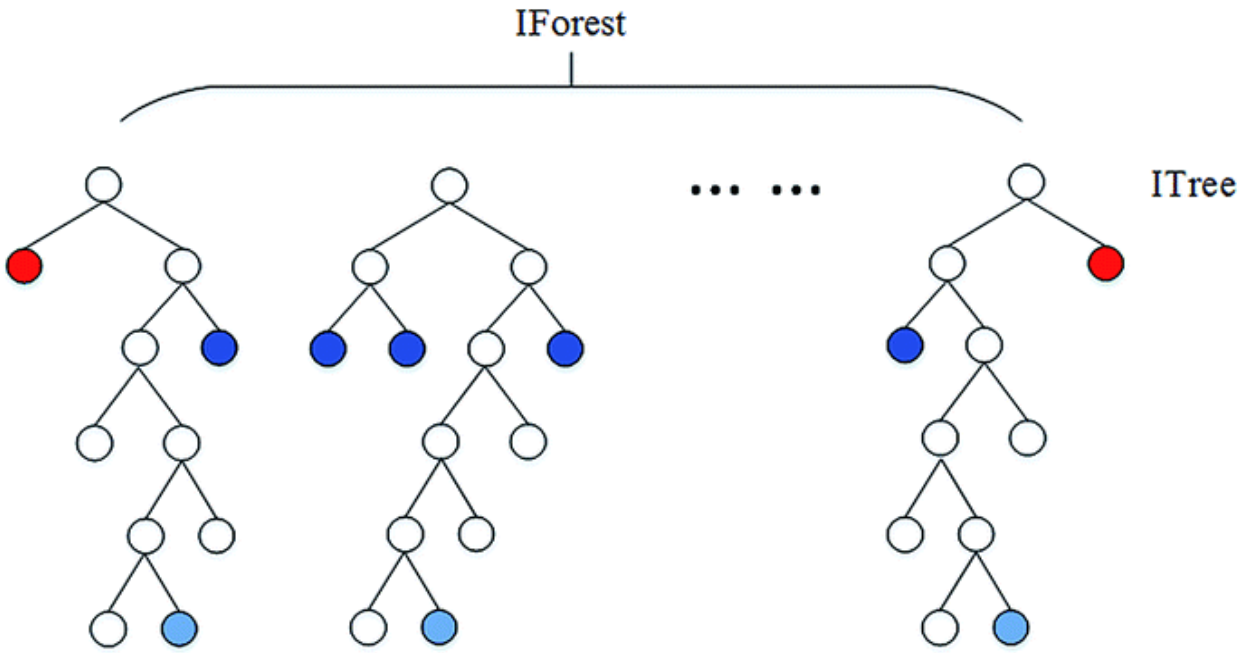


Рисунок 2.2 – Загальна схема виявлення аномалій алгоритмом Isolation Forest

а) Етап навчання

У навчальному етапі ізоляційні дерева будуються шляхом рекурсивного розподілу вказаного навчального набору поки не будуть виділені екземпляри або не буде досягнута конкретна висота дерева, результатом якої є часткова модель. Обмеження висоти дерева автоматично встановлюється під розмір вибірки, яка є приблизно середньою висотою дерева:

$$l = ceiling(log2\ \psi), \qquad (2.1)$$

де l – висота дерева, а ψ – розмір вибірки.

Нижче наведено алгоритм побудови ізоляційного лісу:

Вхідні дані: X – вхідні дані, t – кількість дерев, ψ – вибірки

Вихідні дані: набір ізоляційних дерев у кількості t

1. І**ніціалізація** *Forest*
2. Встановлення ліміту висоти $l = ceiling(log2\ \psi)$
3. **for** i = 1 to t **do**
4. $X' \leftarrow sample(X, \psi)$
5. $Forest \leftarrow Forest \cup iTree(X', 0, l)$
6. **end for**
7. **return** *Forest*

Алгоритм визначення ізоляційних дерев:

Вхідні дані: *X* – вхідні дані, *e* – висота поточного дерева, *l* – ліміт висоти

Вихідні дані: ізоляційне дерево

1. **if** $e \geq l$ *or* $|X| \leq 1$ **then**
2. return *exNode{Size* $\leftarrow |X|$*}*
3. **else**
4. нехай *Q* - перелік атрибутів у *X*
5. випадково обирається атрибут q ∈ Q
6. випадково обирається точка розколу *p* з *max* та *min* значеннями атрибуту *q* в *X*
7. $X_l \leftarrow$ *filter(X, q < p)*
8. $X_r \leftarrow$ *filter(X,* $q \geq p$*)*
9. return *inNode{Left* $\leftarrow$ *iTree(*X_l *, e + 1, l),*
10. *Right* $\leftarrow$ *iTree(*X_r*, e + 1, l),*
11. *SplitAtt* $\leftarrow$ *q,*
12. *SplitValue* $\leftarrow$ *p}*
13. **end if**

б) Етап оцінки

На етапі оцінки показник *s* аномалії виводиться з очікуваної довжини шлях *E(h(x))* для кожного тестового екземпляра. *E(h(x))* отримується шляхом проходження екземпляра через кожен iTree у iForest. Використовуючи функцію PathLength довжина шляху *h(x)* отримується шляхом підрахунку кількості ребер *e* від кореневого вузла до вузла, що закінчується, коли екземпляр *x* проходить через *iTree*. Коли x закінчується на зовнішньому вузлі, де *Size > 1*, повертається значення e з додаванням корегуванння *c(Size).*

$$c(n) = 2H(n - 1) - (2(n - 1)/n), \qquad (2.2)$$

де *H(i)* – гармонічне число, яке можна оцінити за по *ln (i) + 0,5772156649* (константа Ейлера).

Корегування враховує незабудоване піддерево за межею висоти дерева.

Коли *h(x)* отримано для кожного дерева ансамблю, оцінка аномалії виконується за обчисленням *s(x, ψ):*

$$s(x, n) = 2^{-} E(h(x))/c(n) , \qquad (2.3)$$

де *E(h(x))* – середнє значення *h(x)* з колекції ізоляційних дерев.

Складність процесу оцінювання складає

$$O(ntlog\ \psi), \qquad (2.4)$$

де n – розмір даних тестування.

Наведемо алгоритм визначання довжини шляху (*PathLength(x, T, e)*):

Вхідні дані: *x* – екземпляр, *T* - iTree, *e* - поточна довжина шляху;

ініціалізація до нуля при першому виклику

Вихідні дані: довжина шляху x

1. **if** T зовнішній вузол **then**
2. return *e + c(T.size)*
3. **end if**
4. *a* ← *T.splitAtt*
5. **if** x_a < *T.splitValue* **then**
6. return *PathLength(x, T.left, e + 1)*
7. **else** *{*$x_a \geq$ *T.splitValue}*
8. return *PathLength(x, T.right, e + 1)*
9. **end if**

Після того, як отримано всі довжини шляху від екземплярів даних, для виявлення аномалій необхідно просто відсортувати дані за *s(x, ψ)*.

Загальний алгоритм роботи програмного забезпечення продемонстровано на Рисунку 2.3.

2.2.2 Приклад роботи алгоритму

Для кращого розуміння роботи алгоритму наведемо простий та наглядний приклад виявлення аномального повідомлення, для цього нехай розмір Sliding Window має довжину 1 та приймає лише по одному повідомленню за раз.

У потоці даних до часового вікна надійде наступне повідомлення:

«До Києва насувається ураган. Синоптики просять не виходити з дому та сховатися у приміщеннях з 8 вечора до 4 ранку.»

Словник нормалізованих даних, які вже надходили раніше виглядає представлено у Таблиці 2.1.

Таблиця 2.1 – Словник Bag of Words

Слово (id)	Загальна кількість	Кількість в документах	Вага
Київ	350	278	-0,0569
вечір	300	300	0
ранок	200	200	0
синоптики	500	500	0
просити	89	60	-0,0523
ховатися	54	54	-0,0333
насуватися	200	186	-0,0548
вийти	222	170	-0,0664

Після передобробки даних та оновлення словника, його структура представлена у Таблиці 2.2.

Таблиця 2.2 – Словник Bag of Words після оновлення даних

Слово (id)	Загальна кількість	Кількість в документах	Вага
Київ	351	279	-0,0573
вечір	301	301	0
ранок	201	201	0
синоптики	101	101	0
просити	90	61	-0,0475
ховатися	55	55	-0,0314
насуватися	201	187	-0,0552
вийти	223	171	-0,0662
ураган	1	1	-0,004

Розподілення наших даних описується точковою діаграмою розподілення щільності даних, що зображена на Рисунку 2.3.

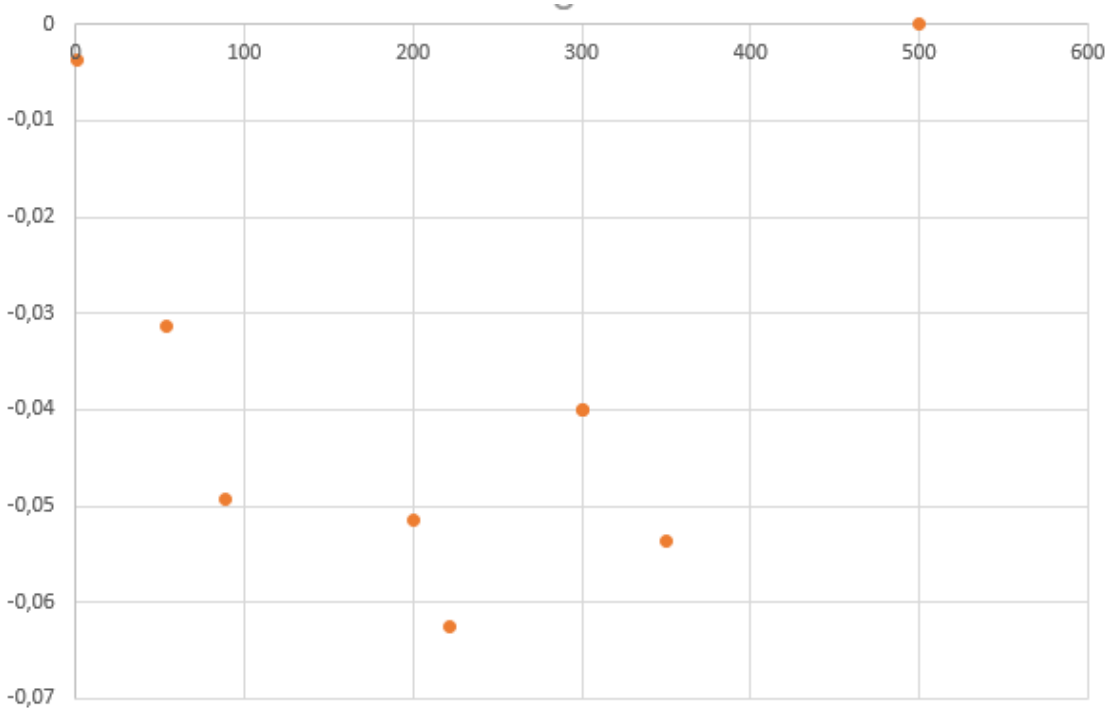


Рисунок 2.3 – Точкова діаграма розподілення тестових даних

Наступник кроком до виявлення аномального вікна є побудова ізоляційного лісу. Наведемо приклад знаходження аномальної точки.

Максимальна довжина нашого дерева становить 3 вузла

Першим кроком є обирання точки розколу. Нехай точкою буде параметр *Загальна кількість* зі значенням розколу 100. Другою точкою розколу нехай буде параметр Вага зі значенням -0,02. Побудоване дерево ізоляції наведено на Рисунку 2.4.

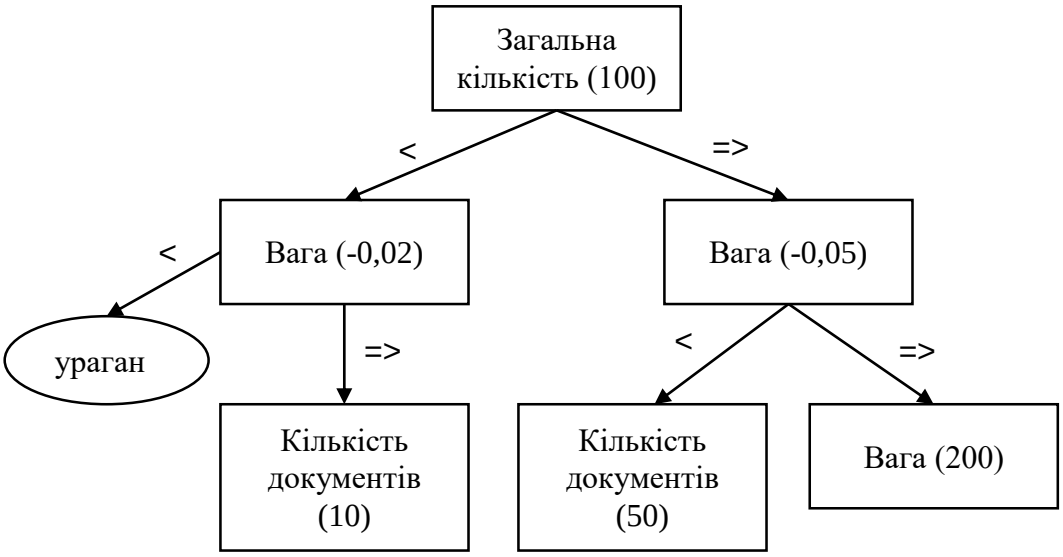


Рисунок 2.4 – Дерево ізоляції для аномалії «ураган»

На точковій діаграмі ізоляцію аномалії можна зобразити наступним чином як на Рисунку 2.5.

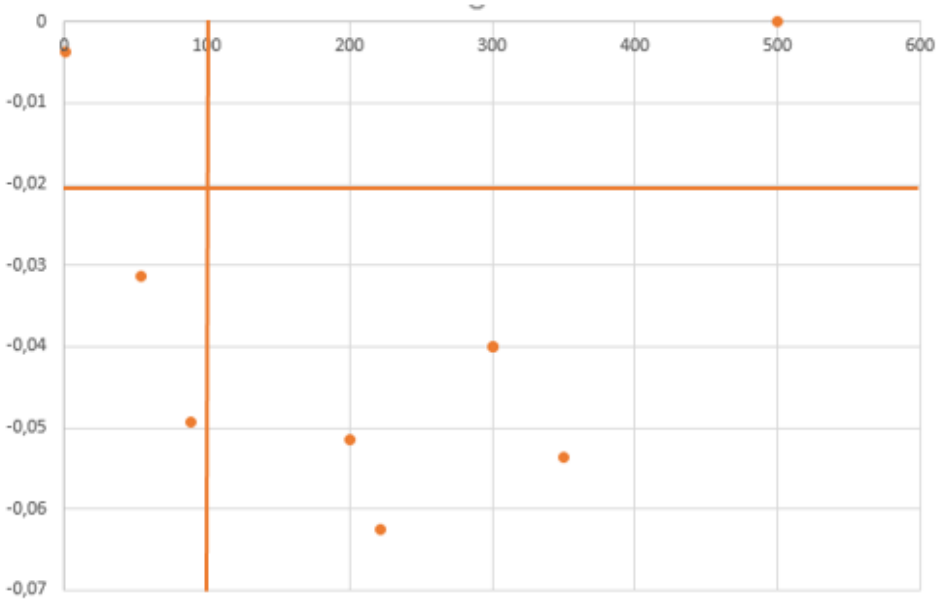


Рисунок 2.5 – Ізоляція аномалії методом Isolation Forest

Побудова дерев продовжується, поки не закінчиться вибірка. Узагальнений алгоритм обробки даних на визначення аномалій можна розглянути на Рисунку 2.6.

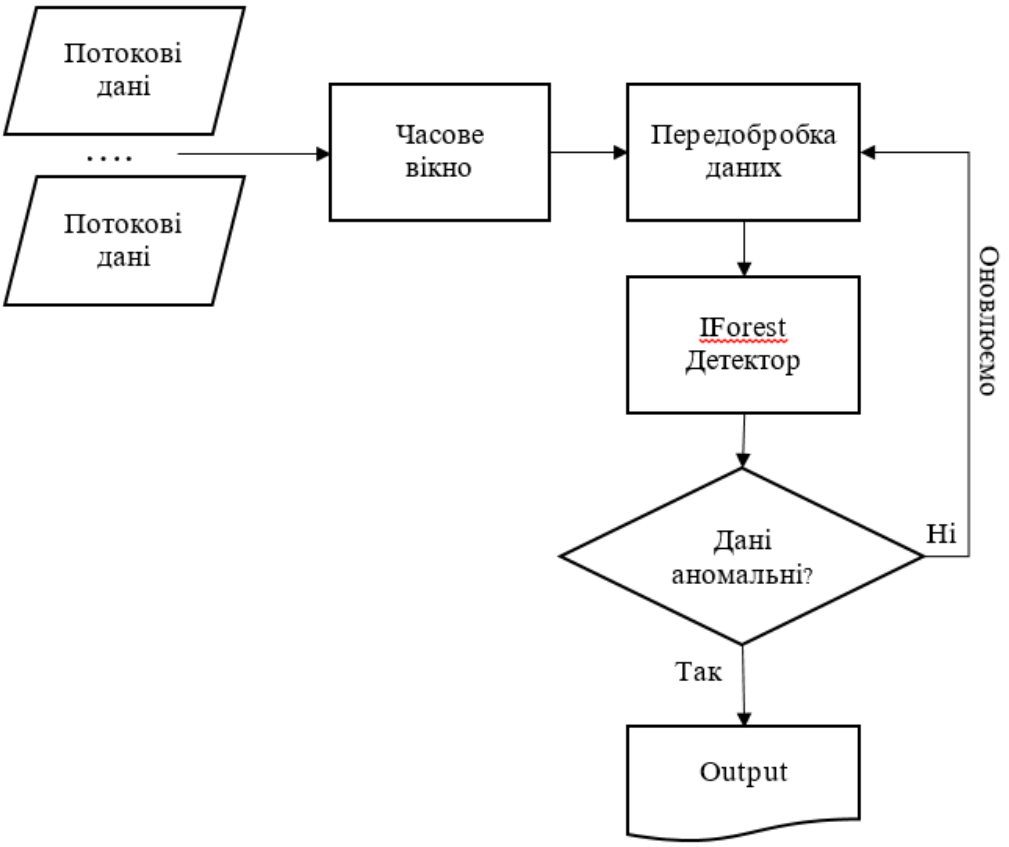


Рисунок 2.6 – Загальний алгоритм виявлення аномалій в потах даних

Висновки до розділу 2

В якості математичного обґрунтування використовується підхід часових вікон та алгоритм ізоляційних дерев для виявлення аномалій.

Sliding Window є швидким та зручним методом для групування при парсингу потокових даних, що дозволяє відслідковувати час находження кожного екземпляру даних та виявляти вікна, у яких дані відсутні.

Алгоритм Isolation Forest є гарним методом вирішення проблеми аналізу потокових текстових даних через невелику складність реалізації, швидку обробку невеликих масивів даних та основний напрям використання алгоритму для пошуку аномалій, завдяки ізолювання кожного елемента у вибірці даних, що потрапляють до Sliding Window.

Детальний алгоритм роботи методу Isolation Forest наведено на графічному матеріалі до диплому.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис стеку розробки

3.1.1 Мова програмування Python

Починаючи з випуску в 1991 році, мова програмування Python [35] надзвичайно популярна і широко використовується в обробці даних. Ось деякі причини такої популярності:

а) об'єктно-орієнтована мова;

б) загального призначення;

в) має багато розширень і велику підтримку спільноти;

г) простий і легка для розуміння і вивчення;

д) має багато бібліотек для обробки потокових та великих.

3.1.2 Фреймворк Apache Spark

Spark [3] – фреймворки для кластерних обчислень і великомасштабної обробки даних. Spark пропонує набір бібліотек на 3 мовах (Java, Scala, Python) для уніфікованого обчислювального движка.

Уніфікований: в Spark немає необхідності збирати програму з декількох API або систем. Spark надає вбудовані API для виконання роботи.

Обчислювальний движок: Spark підтримує завантаження даних з різних файлових систем і виконує в них обчислення, але сам не зберігає ніяких даних постійно. Spark працює виключно в пам'яті, що дає безпрецедентну продуктивність і швидкість.

Фреймворк Spark складається з ряду бібліотек, які створені для вирішення завдань Data Science. Spark включає бібліотеки для SQL (SparkSQL), машинного навчання (MLlib), обробки потокових даних (Spark Streaming і Structured Streaming) та обробки графів (GraphX).

3.1.3 Бібліотеки

а) Pandas

Pandas [36] – бібліотека програмного забезпечення, написана мовою програмування Python для маніпулювання та аналізу даних. Зокрема, він пропонує структури даних та операції для управління числовими таблицями та

часовими рядами. Це безкоштовне програмне забезпечення, випущене за ліцензією BSD. Назва походить від терміну "панельні дані", термін економетрики для наборів даних, що включає спостереження протягом деякого періоду часу за одним і тими же особами.

б) PyMorphy2 [12] – це бібліотека для мови програмування Python, яка призначена для морфологічного аналізу тексту українською та російською мовами.

3.2 Моделювання програмного забезпечення

Користувач системи має наступні можливості у системі:

а) додати посилання на ресурс з потоковими даними;

б) переглянути результати аналізу даних на наявність аномалій.

Розробку програмних модулів можна розділити на наступні кроки:

а) потокове завантаження даних;

б) передобробка даних;

в) побудова лісу ізоляцій;

г) перевірка на наявність аномальних даних.

Загальний алгоритм роботи системи виглядатиме наступним чином:

Вхідні дані: x – потік даних, *t* – проміжок часу для часового вікна

Вихідні дані: екземпляр аномальних даних *y*

1. Виділити Sliding Window;

2. Виконати передобробку даних;

3. Побудувати ліс Ізоляцій;

4. **if** екземляр з Sliding Window є аномальним:

5. Вивести екземпляр аномальних даних;

6. **else** повернутися до пункту 3.

Детальний алгоритм роботи програмного забезпечення наведено на графічному матеріалі диплому.

На Рисунку 3.1 Зображена діаграма використання для користувача з системою.

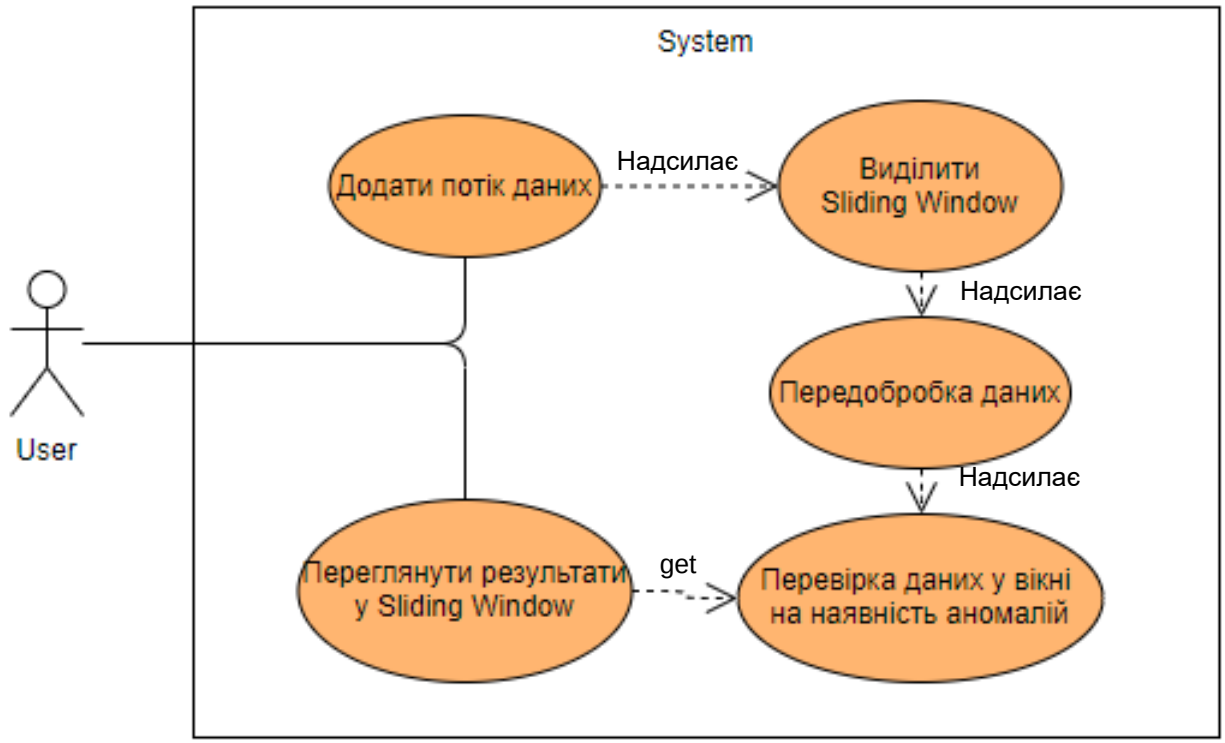


Рисунок 3.1 – Діаграма використання

3.3 Архітектура програмного забезпечення

Розроблювальне математичне та програмне забезпечення складається з наступних модулів:

a) parsing;

b) preprocessing;

c) iforest.

Модуль parsing відповідає за підключення нових потоків даних та отримання даних у реальному часі та розбиття їх на часові вікна.

Модуль preprocessing відповідає за очистку даних, стемінг, лематизацію, створення, поповнення словника та оцінку слів в словнику.

Модуль iforest відповідає за виявлення аномалій та побудову моделі лісу Ізоляцій.

Программе забезпечення працює на основі PySpark і включає в себе компоненти SparkStreaming, Pandas та TwitterApi. Дані збираються з ресурсу Twitter. Результати обробки виводяться в консоль.

Архітектуру програмного забезпечення наведено на на Рисунку 3.2.

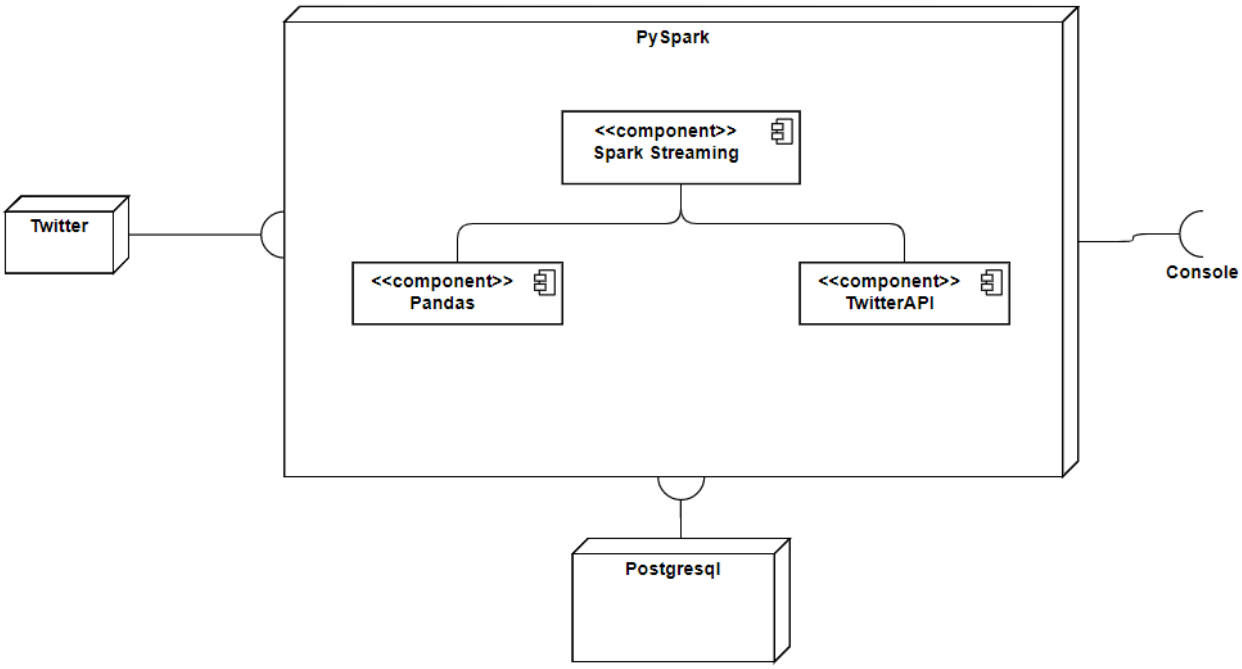


Рисунок 3.2 – Діаграма розгортання програмного забезпечення

Таблиця 3.1 – Опис методів програмного забезпечення

Метод	Призначення	Аргументи
PrD.word_extraction	Очищає текст та приводить всі символи до нижнього реєстру	sentences
PrD.tokenize	Розбиває текстові дані на окремі слова	sentences
PrD.generate_bow	Створює або оновлює словник	allsentences
PrD.lemmatization	Приводить слово до нормалізованого вигляду	sentences
PrD.vectorization	Векторизує тесктові дані за TF-IDF	sentences
ParD.add_data_stream	Задає url ресурсу для парсингу даних	url
ParD.sliding_window	Парсить дані та групує іх по Sliding Window	sentences
AD.isolation_forest	Будує ліс Ізоляцій	allsentence, len(allsentence)
AD.isolation_tree	Будує ізоляційне дерево	X, e, l

Продовження таблиці 3.1

AD.path_length	Обраховує довжину шляху	X, t, e
AD.anomaly_detection	Классифікує дані як аномальні або нормальні	X, lpath
AD.knn		allsentence

База даних включає у собі таблиці:

а) bag-of-words з полями word (str), dtime (datatime), num-of-doc (int), num-of-word (int), weight (float) для збереження словника;

б) anomaly з полями id (int), message (str), dtime (datatime) для збереження результатів аналізу.

Схематичний вигляд бази даних наведений на Рисунку 3.3.

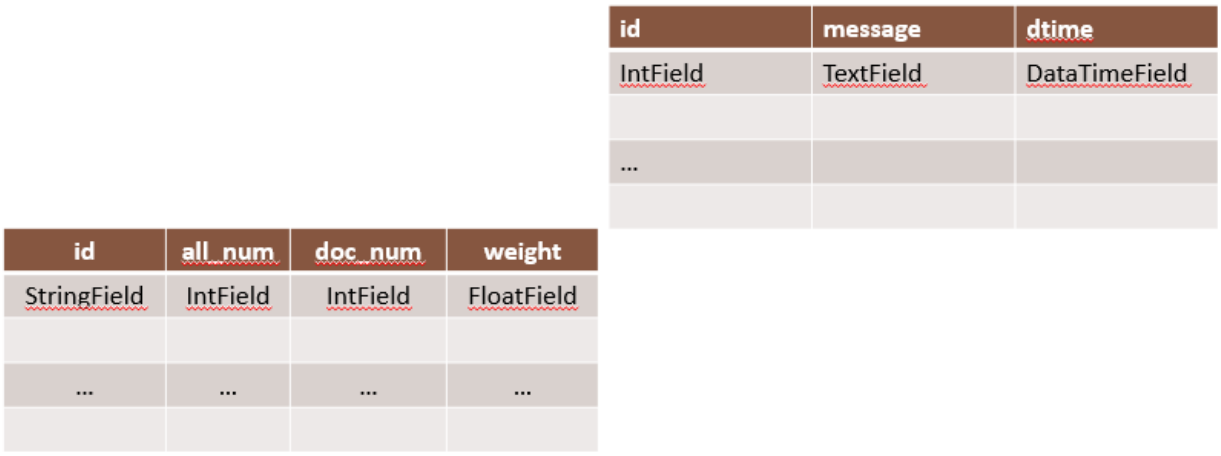


Рисунок 3.3 – Структура бази даних

3.4 Керівництво користувача

Для розгортання додатку необхідно:

а) встановити мову програмування Python версії 3 або вище;

б) встановити пакет pip;

в) завантажити проект до віртуального середовища;

г) у командному рядку віртуального середовища виконати команду *pip install -r requirements.txt;*

д) на вхід до методу add_data_stream(url) подати посилання на потік даних;

е) запустити проект.

Висновки до розділу 3

Для розробки такого програмного забезпечення використовується мова програмування Python, бібліотека для роботи з Big Data pandas та Apache Spark.

Для моделювання програмного забезпечення було використано діаграму діяльності і на її основі побудовано детальний алгоритм роботи програмного забезпечення.

Архітектура програмного забезпечення включає три основні програмні модуля та базу даних з двома таблицями. Для виконання досліджень до програмного забезпечення дописано додатковий метод для алгоритму k-найближчих сусідів knn (allsentences).

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ ВИЯВЛЕННЯ АНОМАЛІЙ В ПОТОКАХ ТЕКСТОВИХ ДАНИХ

4.1 Опис експерименту

Для дослідження ефективності розпізнавання аномалій, порівняємо обраний для розробки алгоритм Isolation Forest з методом k-найближчих сусідів.

Для проведення експерименту згенеровано словник з даних, що були опубліковані за 7 днів з ресурсу Twitter за ключовим словом «Зеленський» і за цим же ключовим словом вручну відібрано 50 україномовних повідомлень, серед яких присутні 17 аномальних, які наведені у Таблиці 5.1.

Таблиця 4.1 – Аномальні повідомлення для аналізу даних

№	*Текст повідомлення*
1	Президент Зеленський присвоїв 45 жінкам з Прикарпаття звання «Мати-героїня» Президент України Володимир Зеленський підписав указ № 893 «Про присвоєння почесного звання «Мати-героїня». Почесне звання отримали 847 українських жінок
2	Зеленський другий. Топ запитів Google у 2019-му
3	Зеленський підписав закон щодо виділення 200 млн грн на будівництво "Охматдит"
4	Париж. Зеленський почав і програв
5	#Президент Естонії Керсті #Кальюлайд пояснила свій подарунок Володимиру #Зеленський
6	Із Парижа Путін повернувся дуже роздратованим, тому що попередньо всі пункти порядку денного було погоджено, але в останній момент, під час особистої зустрічі президент України В. Зеленський за всіма погодженими пунктами порядку денного відмовив.
7	«Охматдит» добудують швидше: Володимир Зеленський підписав закон щодо виділення 200 млн грн з Фонду Президента України — Офіційне інтернет-представництво Президента України:
8	News From People «Охматдит» добудують швидше на 4-5 місяців: Володимир Зеленський підписав закон щодо виділення 200 млн грн з Фонду Президента України.
9	Зеленський підписав закон про виділення 200 млн грн на будівництво лікувально-діагностичного комплексу "Охматдиту"

Продовження таблиці 4.1

10	Зеленським виділено кошти на добудову ще одного блоку в Охматдит.Блок буде оснащений і введений в експлуатацію вже в 2020.Всі фракції проголосували ЗА! От,як не красти, то й справи краще йдуть!
11	200 млн грн. з президентського фонду Зеленського виділять на завершення будівництва лікарні «Охматдит
12	Німецько-український фонд стане майданчиком для програми кредитування бізнесу, яку раніше анонсував президент Володимир Зеленський.
13	Володимир Зеленський на переговорах тримався гідно та достойно, а карлик-терорист явно нервував, перебираючи та підтанцьовуючи весь час ногами під столом...А все тому, що це перший Президент України, на якого в путіна немає компромату
14	Президент України Володимир Зеленський підписав закон щодо роздержавлення спиртової галузі.
15	«Газпром» борг у 3 млрд дол. може сплатити газом. Президент України Володимир Зеленський вважає це компромісним варіантом
16	Спільна прес-конференція лідерів Нормандської четвірки була досить красномовною. Володимир Зеленський був максимально обережним, коли у своїй промові та відповідях журналістів торкався...
17	Президент Володимир Зеленський телефоном привітав Шарля Мішеля зі вступом на нову посаду голови Європейської ради, а також обговорив питання транзиту газу та продовження санкційного тиску на Російську Федерацію

На вхід до алгоритмів дані подаються одним часовим вікном. На виході після обробки алгоритми мають віддати список аномальних повідомлень, після чого буде порівняно точність виявлення аномальних повідомлень між алгоритмами.

4.2 Результати експерименту

Таблиця 4.2 – Порівняння правильних та хибних розпізнавань аномалій

Тип розпізнавання аномалії	Isolation Forest	Метод найближчих сусідів
Правильний	15 (88%)	13 (76,4%)
Хибний	2 (92%)	5 (82%)

Таблиця 4.3 – Нерозпізнані аномальні повідомлення

Isolation Forest	Метод k-найближчих сусідів
Президент Володимир Зеленський телефоном привітав Шарля Мішеля зі вступом на нову посаду голови Європейської ради, а також обговорив питання транзиту газу та продовження санкційного тиску на Російську Федерацію	Президент Володимир Зеленський телефоном привітав Шарля Мішеля зі вступом на нову посаду голови Європейської ради, а також обговорив питання транзиту газу та продовження санкційного тиску на Російську Федерацію
Із Парижа Путін повернувся дуже роздратованим, тому що попередньо всі пункти порядку денного було погоджено, але в останній момент, під час особистої зустрічі президент України В. Зеленський за всіма погодженими пунктами порядку денного відмовив.	«Охматдит» добудують швидше: Володимир Зеленський підписав закон щодо виділення 200 млн грн з Фонду Президента України — Офіційне інтернет-представництво Президента України:
	Президент України Володимир Зеленський підписав закон щодо роздержавлення спиртової галузі.
	Зеленський підписав закон про виділення 200 млн грн на будівництво лікувально-діагностичного комплексу "Охматдиту"

Таблиця 4.4 – Хибні розпізнання звичайних повідомлень як аномальних

Isolation Forest	Метод k-найближчих сусідів
Україна все ж отримає кредит МВФ. Про те, що з Фондом вдалося домовиться по новій трирічній програмі на 5,5 млрд доларів повідомили президент в Facebook і президент Володимир Зеленський, і прем’єр-міністр Олексій Гончарук, назвавши цю новину...	Володимир Зеленський провів телефонну розмову з Президентом Європейської ради — Офіційне інтернет-представництво Президента України:

Продовження таблиці 4.4

Володимир Зеленський розчарований тим, що "партнери" не підтримали його у питанні кордону, хоча домовленості були	Нова істерика від слабких духом-чому Володимир Зеленський не кричав путіну- агресор, вбивця, покидьок ….? Вам полонених додому, чи дешеві понти?Давайте вже дорослішати, бути відповідальними до тих хто...
	Президент України Володимир Зеленський заявив, що запропонував створити в межах Тристоронньої контактної групи у Мінську підгрупу для вирішення питання повернення кордону під український контроль
	Володимир Зеленський: Для мене найголовніша перемога – ми домовилися про звільнення за принципом «всіх на всіх» до 31 грудня.
	Думки Голови ОП/Мысли Главы ОПДекілька новин з Парижу: -1 раунд перемовин був важкий, але оптимістичний - російська сторона усіляко затягувала зустріч президентів один на один, чого дуже хотів Володимир Зеленський

Результати аналізу роботи алгоритмів наведені на Рисунках 4.1-4.4.

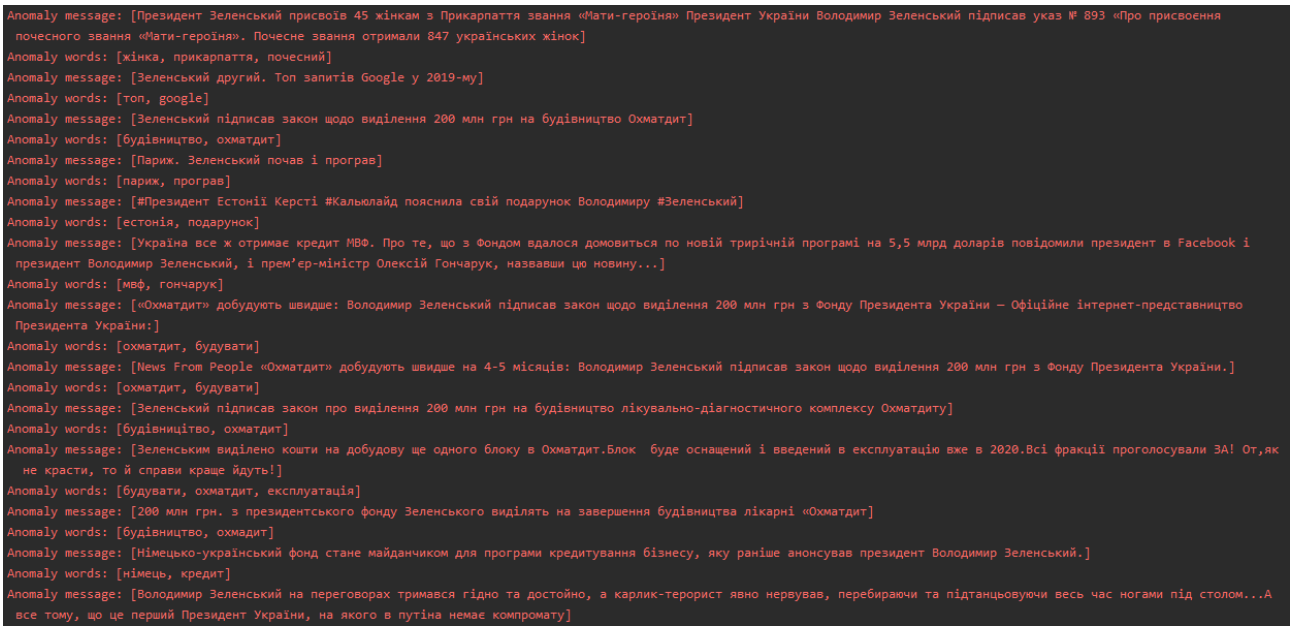


Рисунок 4.1 – Результати виявлення аномалій методом Isolation Forest

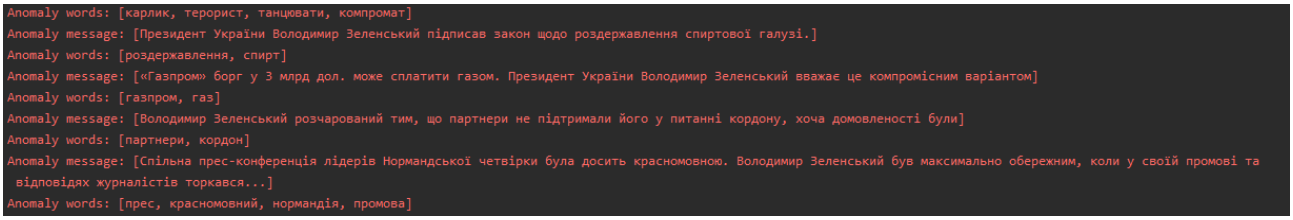


Рисунок 4.2 – Продовження результатів виявлення аномалій методом Isolation Forest

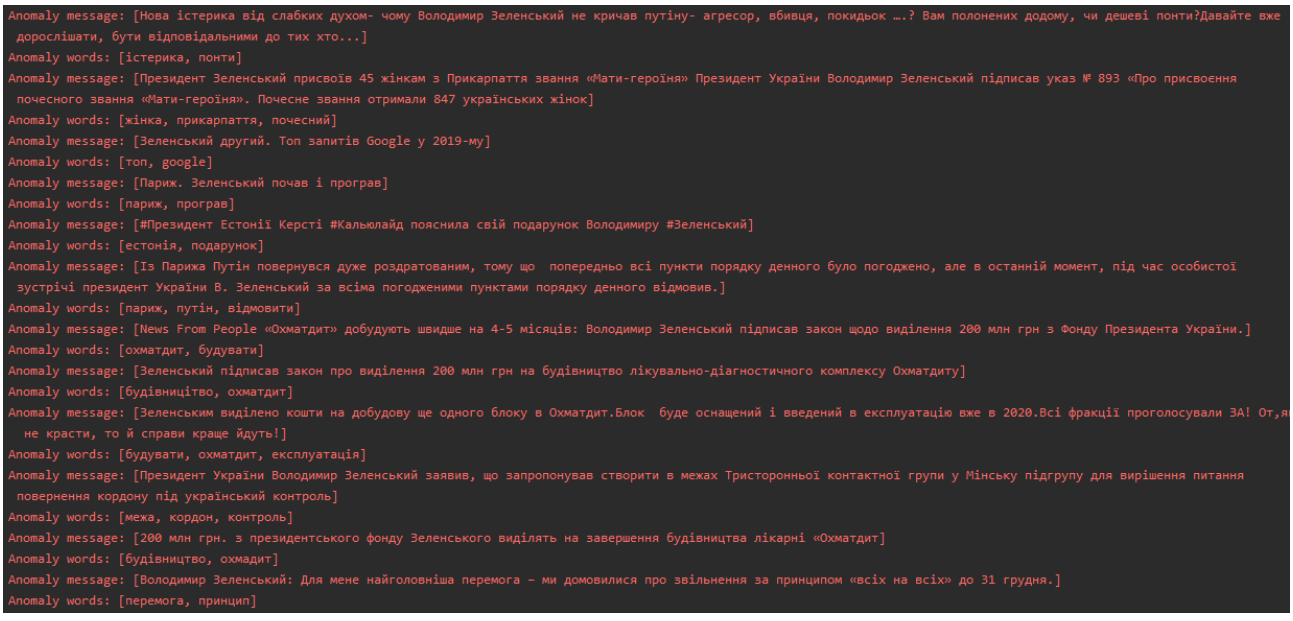


Рисунок 4.3 – Результати виявлення аномалій методом k-найближчих сусідів

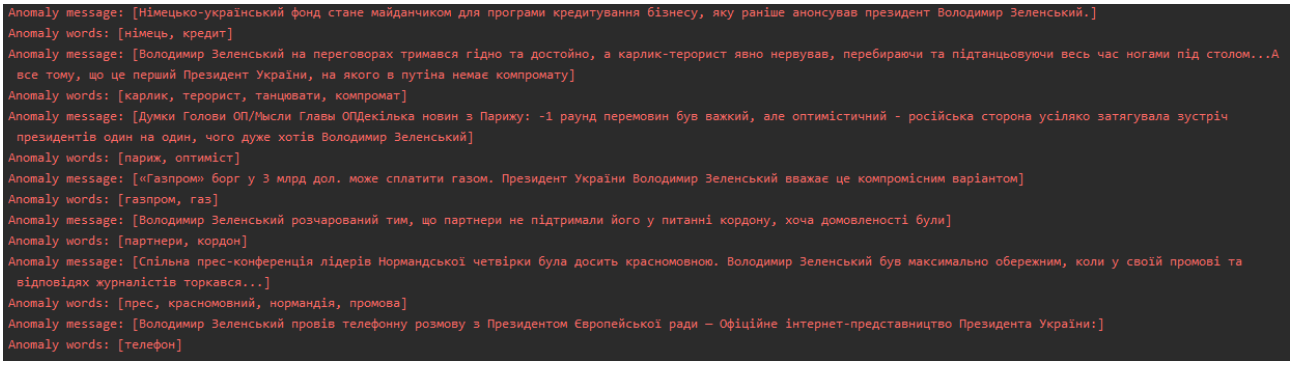


Рисунок 4.4 - Продовження результатів виявлення аномалій методом k-найближчих сусідів

Висновки до розділу 4

В результаті порівняння алгоритмів виявлення аномалій на вибірці даних з ресурсу Twitter було виявлено перевагу в виявлені аномальних даних алгоритмом Isolation Forest над методом k-найближчих сусідів.

Алгоритм Isolation Forest зробив менше похибок у виявленні аномальних даних і на даному випадку точність виявлення аномалій склала 88%, а алгоритму k-найближчих сусідів 76,4%.

Похибка на всьому потоці даних склала 8% і 18% відповідно.

З цих даних можна скласти висновок, що для виявлення аномалій на потокових даних алгоритм Isolation Forest використовувати більш ефективно ніж метод k-найближчих сусідів.

5 РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

5.1 Опис ідеї

Таблиця 5.1 – Опис ідеї стартап проекту

Зміст ідеї	*Напрямки застосування*	*Вигоди для користувача*
Розробити математичне та програмне забезпечення для виявлення аномалій в україномовних потокових даних. Надати користувачу можливість відслідковувати аномалії та переглядати інформацію про них.	Виявлення «гарячих» трендів у медіа.	Можливість першими дізнаватися про набираючі популярність теми.
	Виявлення виявлення крапкових аномалій.	Можливість виявити у потоці даних аномальні дані, які не набирають популярність.
	Виявлення аномалій в режимі реального часу.	Можливість дізнаватися про виявлення аномалій 24/7.
	Підтримка української мови	У конкурентів не реалізована можливість виконувати морфологічний аналіз даних українською мовою.

Таблиця 5.2 – Визначення сильних слабких та нейтральних характеристик ідеї проекту

№ п/п	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепції конкурентів*				*W (слабка сторона)*	*N (нейтральна сторона)*	*S (сильна сторона)*
		Розроблене ПЗ	*Numenta*	*Avora*	*Splunk Enterprise*			
1.	Доступність для користувачів	Безкоштовно	Безкоштовно	Платна підписка (120$ на рік)	Платна підписка (80$ на рік)			+

Продовження таблиці 5.2

2.	Якість	Вища середнього	Висока	Висока	Середня		+	
3.	Повнота функціоналу	Весь функціонал	Відсутня підтримка української мови	Відсутня можливість аналізувати текстові дані	Не працює з потоковими даними			+
4.	Зручність інтерфейсу	Висока	Висока	Виксока	Висока		+	
5.	Витрати на розробку та впровадження	Малі	Великі	Середні	Малі	+		

5.2 Технологічний аудит проекту

Таблиця 5.3 – Технологічна здійсненність проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Розробка програмного забезпечення	Python	Наявна	Доступна
		PySpark	Наявна	Доступна
		Pymorphy2	Наявна	Доступна
2	Розробка бази даних	Hadoop	Наявна	Доступна
3	Розробка алгоритму для виявлення аномалій	Алгоритм Isolation Forest для виявлення аномалій в потокових даних	Необхідно розробити	Доступна, потребується додаткове вивчення та допрацювання алгоритму
4	Візуалізація	ploty	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Обрано технологічний стек Python, PySpark та Hadoop як кращі серед наявних на ринку та доступних для членів команди. Також будуть використовуватися алгоритми пошуку аномалій та візуалізація даних.				

5.3 Аналіз ринкових можливостей запуску стартап проекту

Таблиця 5.4 – Попередня характеристика ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількість головних гравців, од	13
2	Загальний обсяг продаж, грн/ум.од	2 785 000 000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Юридичні обмеження
5	Специфічні вимоги до стандартизації та сертифікації	Вимоги до програмного забезпечення
6	Середня норма рентабельності в галузі (або по ринку), %	42 %

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Виявлення набираючих популярність тем	Люди з щільним графіком, які не мають час на переглядання повної стрічки новин	відсутні	Інтуітивна зрозумілість в користуванні
2	Виявлення аномальних даних в україномовних даних для бізнес рішень	Бізнес компанії, які хочуть або вже працюють з українським ринком медіа або науки	відсутні	Можливість інтегрувати API до своєї системи
3	Виявлення аномальних даних в україномовних даних для наукових досліджень	Наукові установи та науковці, які займаються аналізом даних	відсутні	Інтуітивна зрозумілість в користуванні та можливість використовувати API у наукових цілях

Продовження таблиці 5.5

4	Виявлення аномальних даних в україномовних даних для подальшої розробки програмного забезпечення	Розробники програмного забезпечення для аналізу даних	відсутні	Можливість інтегрувати API до своєї системи

Таблиця 5.6 – Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Незацікавленість компаній у програмному забезпеченні	Через деяку відсталість українського бізнесу, що працює на національному ринку, компанії можуть виявитися незацікавленими в продукті, неокупність проекту	Розширення кількості мов для аналізу тексту, та спроба вийти на сусідні більш розвинені національні ринки. Додати морфологічний аналіз англійської мови та співпрацювати з українським бізнесом, що працює з аутсорсом
2	Непопулярність проекту	Неокупність проекту	Замовити кількісні та якісні дослідження. За їх результатом оптимізувати програмне забезпечення та розробити успішну рекламну компанію

Таблиця 5.7 – Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Високий попит на програмне забезпечення	Окупність проекту	Подальший розвиток продукту
2	Отримання грантів на розвиток продукту	Окупність проекту	Подальший розвиток продукту

Продовження таблиці 5.7

3	Популярність продукту серед розробників	Сильна спільнота, що постійно допомагає покращити проект	Отримання постійного відгуку та правильний напрямок розвитку продукту
4	Отримання підтримки інвесторів	Отримання фінансування для розширення проекту	Збільшення команди, розширення функціоналу проекту

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - олігополія	Існують подібні застосунки, що мають частковий функціонал, але повноцінного конкурента на національному ринку немає	Для початкового старту розробити один якісний алгоритм для виявлення аномалій в україномовних даних. Потім можна розробити ще кілька алгоритмів для виявлення аномалій та розширювати функціонал у інших галузях аналізу текстових даних
2. За рівнем конкурентної боротьби - національний	Можливо існують конкуренти в інших країнах, але наш вектор - Україна	Національний ринок не має подібного програмного продукту
3. За галузевою ознакою - зовнішньогалузева	Може використовуватися як в бізнесі, так і у науковій галузі	Зробити якісний продукт, який буде цікавий до співпраці якомога більшій кількості бізнесу та науковим галузям
4. Конкуренція за видами товарів: товарно-видова	Призначення – аналіз тексту. Відмінність у параметрах – різні мови для аналізу потоків даних	Якість програмного забезпечення та розрекламованість

Продовження таблиці 5.8

5. За характером конкурентних переваг - нецінова	Користувачі цінують якість та зручність користування та швидкість аналізу і актуальність інформації	Провести детальне дослідження ринку, залучити якомога більше клієнтів
6. За інтенсивністю - марочна	Для реклами потрібен бренд	Пріоритет – завоювати довіру користувачів, через розробку якісного продукту. Взаємодія з користувачами через відгуки та GitHub.

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*
Складові аналізу	*Numeta*	*Розробка функціоналу, юридичні питання, відсутність клієнтів*	*Відсутні*	*Кількість бізнес клієнтів, які платять за підтримку функціоналу їх системі*	*Часткова заміна, вища якість, кращі умови*
Висновки	Не інтенсивна	- є можливості входу в ринок - є потенційні конкуренти у інших країнах	-	Якість ПЗ, зацікавленість в користуванні	Обмеження для роботи на ринку через проекти замінники відсутні станом на 2019 рік

Проект може вийти на український ринок, та бути конкурентоспроможним. Для цього потрібно:

а) вигідні умови для співпраці з бізнес-клієнтами;

б) додавати новий функціонал;

в) забезпечувати якісну інтеграцію з іншими системами.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
1	Вигідні умови співпраці для національного бізнесу	Умови співпраці важливі для співпраці з клієнтами
2	Висока якість застосунку	Програмне забезпечення має легко інтегруватися до інших систем, мати зручний інтерфейс
3	Продумана рекламна компанія	Важливий чинник для приваблення клієнтів та спільноти
4	Швидка робота алгоритму та безперебійне завантаження даних	Важливо, щоб клієнти були задоволені швидкістю отримання даних про аномалії

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін ПЗ

№ п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з розроблювальним ПЗ*						
			–3	*–2*	*–1*	*0*	*+1*	*+2*	*+3*
1	вигідні умови співпраці для національного бізнесу		-3						
2	висока якість застосунку						+1		
3	продумана рекламна компанія					0			
4	швидка робота алгоритму та безперебійне завантаження даних				-1				

Таблиця 5.12 – SWOT-аналіз стартап проекту

Сильні сторони: - Підтримка української мови - Унікальний продукт для українського ринку	Слабкі сторони: - Високі ризики неокупності - Вузька направленість

Продовження таблиці 5.12

Можливості: - Розвиток галузі роботи з даними - Великий попит серед українських медіа та соціологічних компаній - Зростання попиту на аналіз даних українською мовою	Загрози: - Обмеженість зацікавлених клієнтів - Не готовність ринку до продукту - Не готовність клієнтів платити за проект

Таблиця 5.13 – Альтернативи ринкового впровадження стартап проекту

№ п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
1	Фокусування на обробці англомовних даних	Висока	1 рік
2	Розширення функціоналу додатку та введення підтримки української мови	Середня	2 роки

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 – Вибір цільових груп потенційних користувачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Розробники, які мають причетність до Data Scince та NLP	Готові	30%	Не інтенсивна	Середня
2	Бізнез, який працює з великою кількістю україномовних даних	Готові	80%	Не інтенсивна	Складно

Продовження таблиці 5.14

3	Наукові установи, що досліджують NLP та Data Since	Готові	100%	Не інтенсивна	Просто
4	Студенти IT факультетів	Готові	20%	Не інтенсивна	Просто
Які цільові групи обрано: бізнес, який працює з великою кількістю україномовних даних, та наукові установи, що досліджують NLP та Data Since.					

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку*
1	Зосередження на одному сегменті (рішення для бізнесу)	Захоплення ринку невеликим темпом, розширюючи аудиторію.	Обрана цільова група, яка принесе максимум прибутку.	Стратегія спеціаліза ції

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
1	Так. Існують інші рішення такої проблеми, але жоден з них не дає підтримку україномовних даних	І нові користувачі і користувачі конкурентів	Ні	Стратегія лідера

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
1	Виявлення аномалій для україномовних даних	Використати бібліотеку pymorphy2	Підтримка української мови	Унікальність
2	Отримувати дані 24/7	Розгортання фреймфорку на Apache Spark	Забезпечення швидкої обробки та неперебійності ПЗ	Відмовостійкість
3	Доступність	Невелика ціна для бізнесу, легка інтеграція	Період безкоштовного користування усіма функціями	Доступність

5.5 Розроблення маркетингової програми стартап проекту

Таблиця 5.18 – Визначення потенційних переваг

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
1	Можливість аналізувати україномовні дані	Морфологічний аналізатор для української мови	У жодного з конкурентів немає аналізу україномовних даних
2	Виявлення аномалій в потокових даних	Аномалії виявляються 24/7 в потокових даних	Система може працювати цілодобово та постійно аналізувати дані
3	Збереження аномальних даних	Система зберігає усі аномальні дані	Система зберігає аномальні дані і надає до них доступ у будь-який час

Таблиця 5.19 – Опис трьох рівнів моделі товару

<table>
<tr><td>Рівні товару</td><td colspan="3">Сутність та складові</td></tr>
<tr><td>І. Товар за задумом</td><td colspan="3">Програмне забезпечення для пошуку аномалій в потоках текстових даних з морфологічним аналізатором української мови</td></tr>
<tr><td rowspan="4">ІІ. Товар у реальному виконанні</td><td>Властивості/характеристики</td><td>М/Нм</td><td>Вр/Тх /Тл/Е/Ор</td></tr>
<tr><td>1. Підтримка української мови
2. Обробка потокових даних
3. Виявлення аномалій</td><td></td><td></td></tr>
<tr><td colspan="3">Якість: математичне та програмне забезпечення</td></tr>
<tr><td colspan="3">Марка: НТУУ КПІ ім. Сікорського</td></tr>
<tr><td rowspan="2">ІІІ. Товар із підкріпленням</td><td colspan="3">До продажу: якість</td></tr>
<tr><td colspan="3">Після продажу: технічна підтримка</td></tr>
<tr><td colspan="4">За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної власності</td></tr>
</table>

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
1	-	80-120 дол на рік	80 000 грн / міс і вище	500-1500 дол на рік для бізнес клієнтів та безкоштовно для некомерційного використання

Таблиця 5.21 – Формування системи збуту

№ п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
1	Орієнтація на безкоштовний використання для некомерції	1. Підтримка системи для бізнесу 1500$ на рік	Середня	Безпосередня

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
1	Орієнтація на якісне програмне забезпечення, яке легко використовувати в своїх системах	IT блогери та IT медіа	Співвідношення ціна/якість	Якість продукту та рекомендацій	Збільшуй свої продажі завдяки новітнім технологіям

Висновки до розділу 5

Був проведений аналіз ринку, визначено плюси та мінуси програмного забезпечення порівняно з конкурентами. Розрахована імовірність успішності впровадження проекту, прораховані ризики та можливості життя проекту після його реалізації.

Можливість зробити програмне забезпечення комерційним можливо завдяки інтеграції його до інших систем та підтримки його роботи з цими системами. Також існують непогані шанси його впровадження завдяки унікальному функціоналу.

ВИСНОВКИ

В даній магістерській дисертації було розроблено та реалізовано програмне забезпечення для виявлення аномалій в потокових текстових україномовних даних за допомогою алгоритму Isolation Forest. Програмний продукт виконаний на мові програмування Python, за допомогою фреймворка PySpark та бібліотек pymorphy2, pandas та Twitter API для парсингу україномовних твітів.

У першому розділі дисертації було проведено аналіз існуючих методів, таких як класифікація, кластеризація, статистичний аналіз, гібридні методи та інші. Проаналізовані методи векторизації текстових даних та способи зберігання словника, для збереження інформації про вже проаналізовані дані. Розглянуті інструменти щодо морфологічного аналізу для даних українською мовою. Описані методи класифікації аномалій сучасними алгоритмами та підходами.

За підсумком аналізу існуючих методів для реалізації програмного забезпечення було обрано алгоритм Isolation Forest для вирішення поставленої задачі.

У другому розділі була описана математична модель алгоритму та сам алгоритм. Для більшого розуміння роботи Ізоляційного Лісу було приведено приклад розбору текстового повідомлення з виявленням аномального слова та графічно проілюстровано цей процес для більшої наглядності. Після опису алгоритму та наведення математичної моделі розроблено загальний та детальний алгоритм роботи програмного забезпечення.

Наступним етапом розробки було змодельоване програмне забезпечення діаграми діяльності та опису загальної послідовності роботи програми та розроблено архітектуру програмного забезпечення, яка включає в себе наступні три модулі: парсинг даних, передобробка тексту та детектор аномалій. Також для зберігання аномальних даних та метрик словника було розроблено невелику базу даних, що включає в собі дві таблиці.

Щоб визначити ефективність роботи алгоритму, було обрано порівняння його з алгоритмом як k-найближчих сусідів. Так як на невеликих вибірках даних

дуже складно дослідити ефективність, а дані які надходять до алгоритму не марковані як аномальні, чи не аномальні. Було створено вибірку, в якій відомо, які дані є аномальними та порівняно результати аналізу двох алгоритмів.

За результатами дослідження ефективності загальна ефективність алгоритму Isolation Forest склала 92%, що на 10% краще за конкурентний алгоритм.

Наукова новизна розробки полягає у розробці адаптованого методу Isolation Forest виявлення аномалій в потоках текстових даних з підтримкою української мови.

Усі поставленні задачі наукової роботи були виконані. Було покращено виявлення аномальних даних в потокових україномовних текстових даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Ilkay A., Amarnath G.: Working with Streaming Data [Electronic Resource] / Ilkay A., Amarnath G. // Big Data Modeling and Management Systems – Electronic Data. – [San Diego: University of California San Diego with Electronic Resource Coursera, 2012] – Mode of access: World Wide Web: https://www.coursera.org/learn/big-data-management/home/week/4 (viewed on September 13, 2019). – Title from the screen.

2) Amazon Web Services, Inc: Amazon Kinesis [Electronic Resource] // Amazon Kinesis – Mode of access: World Wide Web: https://aws.amazon.com/ru/kinesis/ (viewed on September 13, 2019). – Title from the screen.

3) The Apache Software Foundation, Licensed under the Apache License, Version 2.0. : Apache Software Foundation [Electronic Resource] // Apache Spark – Mode of access: World Wide Web: https://www.apache.org/ (viewed on September 13, 2019). – Title from the screen.

4) The Apache Software Foundation, Licensed under the Apache License : Apache Storm [Electronic Resource] // Apache Storm – Mode of access: World Wide Web: https://storm.apache.org/ (viewed on September 13, 2019). – Title from the screen.

5) The Apache Software Foundation, Licensed under the Apache License: Apache Flink [Electronic Resource] // Apache Flink – Mode of access: World Wide Web: https://flink.apache.org/ (viewed on September 13, 2019). – Title from the screen.

6) The Apache Software Foundation, Licensed under the Apache License, Version 2.0. : Apache Spark [Electronic Resource] // Spark Streaming – Mode of access: World Wide Web: https://spark.apache.org/ (viewed on September 13, 2019). – Title from the screen.

7) The Apache Software Foundation, Licensed under the Apache License: Apache Samza [Electronic Resource] // Apache Samza – Mode of access: World Wide

Web: http://samza.apache.org/ (viewed on September 13, 2019). – Title from the screen.

8) Advanced Software Products Group: The Three States of Digital Data [Electronic Resource]: – Mode of access: World Wide Web: http://aspg.com/three-states-digital-data/#.Xej2t-gza00 (viewed on September 14, 2019). – Title from the screen.

9) Facebook [Electronic Resource]: We believe in the potential of people when they can come together – Mode of access: World Wide Web: https://about.fb.com/ (viewed on September 14, 2019). – Title from the screen.

10) Twitter [Electronic Resource]: https://about.twitter.com/ Mode of access: World Wide Web: https://about.fb.com/ (viewed on September 14, 2019). – Title from the screen.

11) Ying Lin: 10 Twitter Statistics Every Marketer Should Know in 2019 [Electronic Resource] – : 10 Twitter Statistics Every Marketer Should Know in 2019 [Infographic] Mode of access: World Wide Web: https://www.oberlo.com/blog/twitter-statistics (viewed on September 14, 2019). – Title from the screen.

12) Korobov M.: Morphological Analyzer and Generator for Russian and Ukrainian Languages [Electronic Resource] – Mode of access: World Wide Web: https://arxiv.org/pdf/1503.07283v1.pdf (viewed on September 16, 2019). – Title from the screen.

13) Open Corpora [Electronic Resource] – Mode of access: World Wide Web: http://opencorpora.org/ (viewed on September 16, 2019). – Title from the screen.

14) Language Tool [Electronic Resource] – Mode of access: World Wide Web: https://languagetool.org/ (viewed on September 16, 2019). – Title from the screen.

15) Олег Бунін: Як вирішити 90% задач з NLP [Electronic Resource] – Mode of access: World Wide Web: https://habr.com/ru/company/oleg-bunin/blog/352614/ (viewed on September 16, 2019). – Title from the screen.

16) Frakes, W. B. Stemming algorithms, Information retrieval – 1992 – data structures and algorithms – Upper Saddle River – NJ: Prentice-Hall, Inc.

17) Saurav Jain: Introduction to Stemming – [GeeksForGeeks: a computer science portal for geeks] – Mode of access: World Wide Web: https://www.geeksforgeeks.org/introduction-to-stemming/ (viewed on September 16, 2019). – Title from the screen.

18) Jocelyn D'Souza: An Introduction to Bag-of-Words in NLP [Electronic Resource] – Mode of access: World Wide Web: https://medium.com/greyatom/an-introduction-to-bag-of-words-in-nlp-ac967d43b428 (viewed on September 16, 2019). – Title from the screen.

19) H. Wu and R. Luk and K. Wong and K. Kwok: "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.

20) Akash Panchal: Text Summarization using NLTK: TF-IDF Algorithm [Electronic Resource] – Mode of access: World Wide Web: https://towardsdatascience.com/text-summarization-using-tf-idf-e64a0644ace3 (viewed on September 16, 2019). – Title from the screen.

21) Jiawei Han and Micheline Kamber and Jian Pei: Data Mining: Concepts and Techniques (3rd) – ISBN: 9780123814791 (2011)

22) В.П. Шкодырев, К.И. Ягафаров, В.А. Баштовенко, Е.Э. Ильина: Обзор методов обнаружения аномалий [Електроний ресурс] - Режим доступу: http://ceur-ws.org/Vol-1864/paper_33.pdf

23) Hodge, V. and Austin, J., 2004. A survey of outlier detection methodologies. Artificial intelligence review, 22(2), pp.85-126.

24) Angiulli, F. and Pizzuti, C., 2002, August. Fast outlier detection in high dimensional spaces. In European Conference on Principles of Data Mining and Knowledge Discovery, pp. 15-27.

25) Kannan, R., Woo, H., Aggarwal, C.C. and Park, H., 2017, June. Outlier detection for text data. In Proceedings of the 2017 SIAM International Conference on Data Mining, pp. 489-497. Society for Industrial and Applied Mathematics.

26) Ramaswamy, S., Rastogi, R. and Shim, K., 2000, May. Efficient algorithms for mining outliers from large data sets. ACM SIGMOD Record, 29(2), pp. 427-438.

27) Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. ACM computing surveys , 41(3), p.15.

28) Chawla, S. and Chandola, V., 2011, Anomaly Detection: A Tutorial. Tutorial at ICDM 2011.

29) Top 10 anomaly detection Software [Electronic Resource] – Mode of access: World Wide Web: https://www.predictiveanalyticstoday.com/top-anomaly-detection-software/ (viewed on September 20, 2019). – Title from the screen.

30) Numeta [Electronic Resource] – Mode of access: World Wide Web: https://numenta.com/ (viewed on September 20, 2019). – Title from the screen.

31) Rapid Miner [Electronic Resource] – Mode of access: World Wide Web: https://rapidminer.com/

32) Zhiguo Ding, MinruiFei: An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window. IFAC Proceedings Volumes. Volume 46, Issue 20, 2013, Pages 12-17.

33) Liu, F.T., Ting, K.M. and Zhou, Z.H., 2008, December. Isolation forest. In International Conference on Data Mining, pp. 413-422. IEEE.

34) Tomashevskii, V.M., Oliynik, Y.O., Yaskov, V.V., Romanchuk, V.M.: Realtime text stream anomalies analysis system. Visnyk of Kherson National Technical University, vol. 66, no. 3, pp. 361–366 (2018)

35) Python [Електронний ресурс] – https://www.python.org/

36) Pandas [Електронний ресурс] – https://pandas.pydata.org/

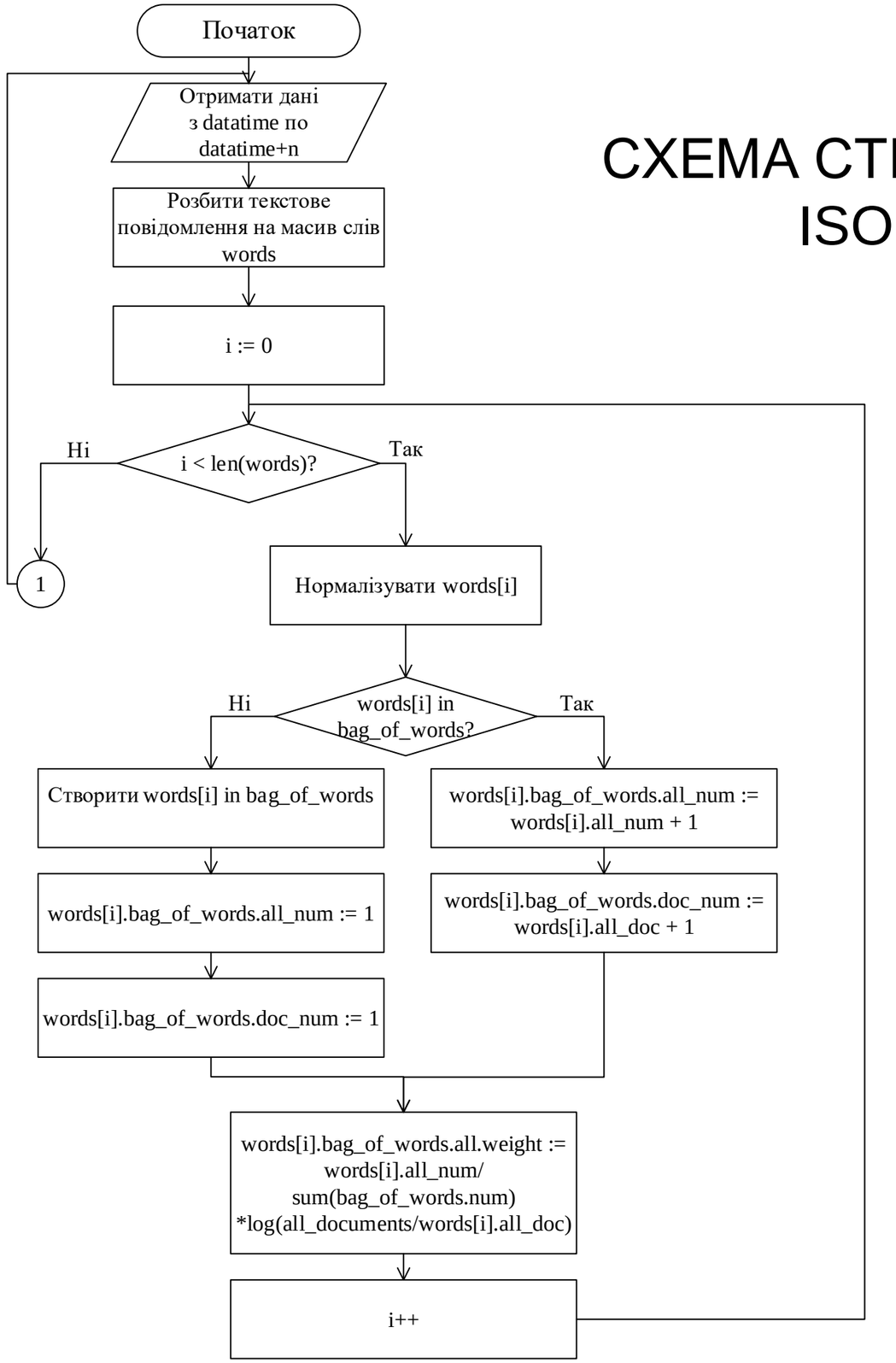
37) Lakshay Arora: An Awesome Tutorial to Learn Outlier Detection in Python using PyOD Library [Електронний ресурс] – Mode of access: World Wide Web: https://www.analyticsvidhya.com/blog/2019/02/outlier-detection-python-pyod/

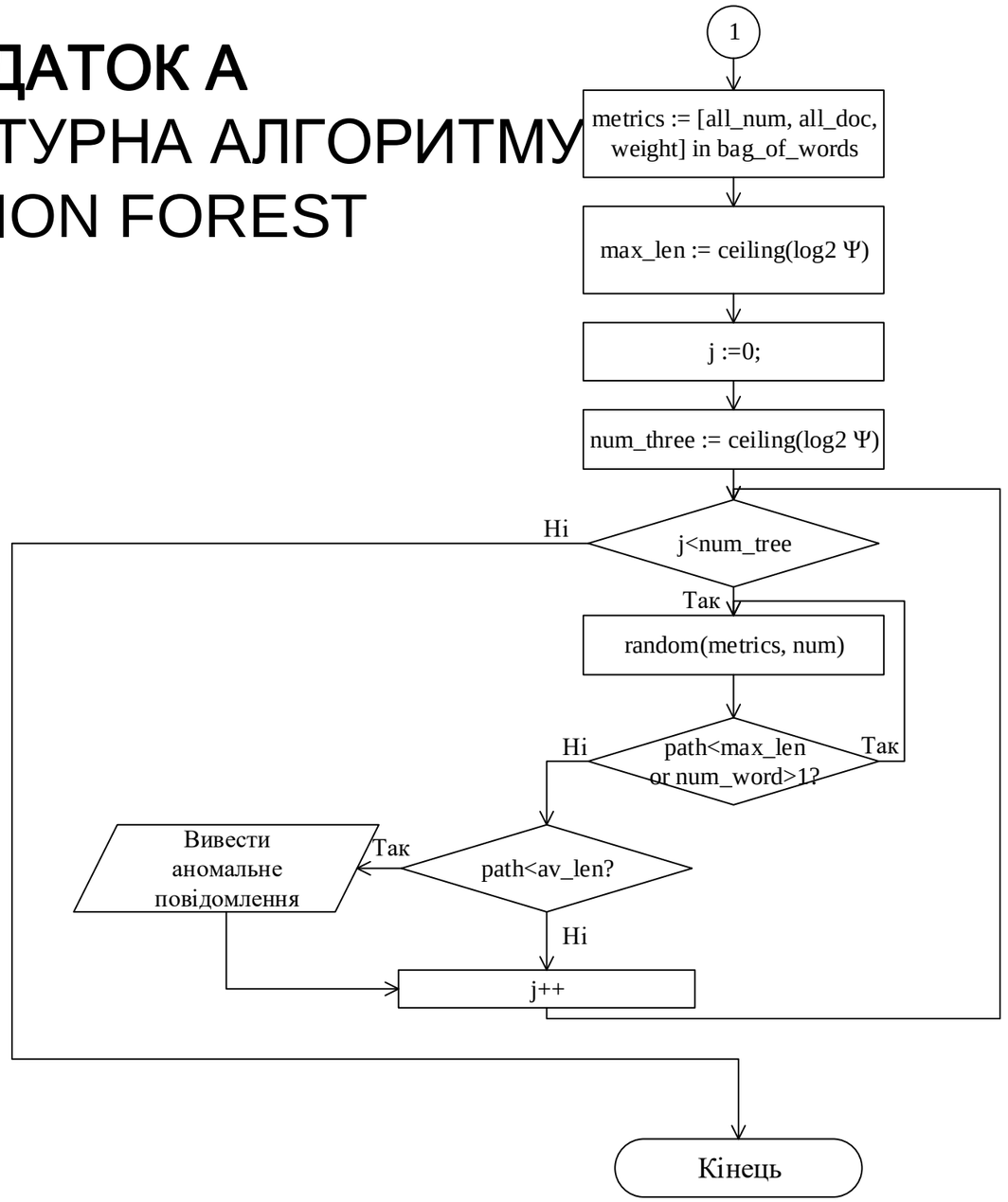
38) Practice Problem: Big Mart Sales III [Dataset] – Mode of access: World Wide Web: https://datahack.analyticsvidhya.com/contest/practice-problem-big-mart-sales-iii/?utm_source=outlierdetectionpyod&utm_medium=blog

39) О.Є. Афанасьєва, Ю.О. Олійник // Матеріали III всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні

системи та технології управління» (ІСТУ-2019) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р.

ДОДАТОК А
СХЕМА СТРУКТУРНА АЛГОРИТМУ ISOLATION FOREST





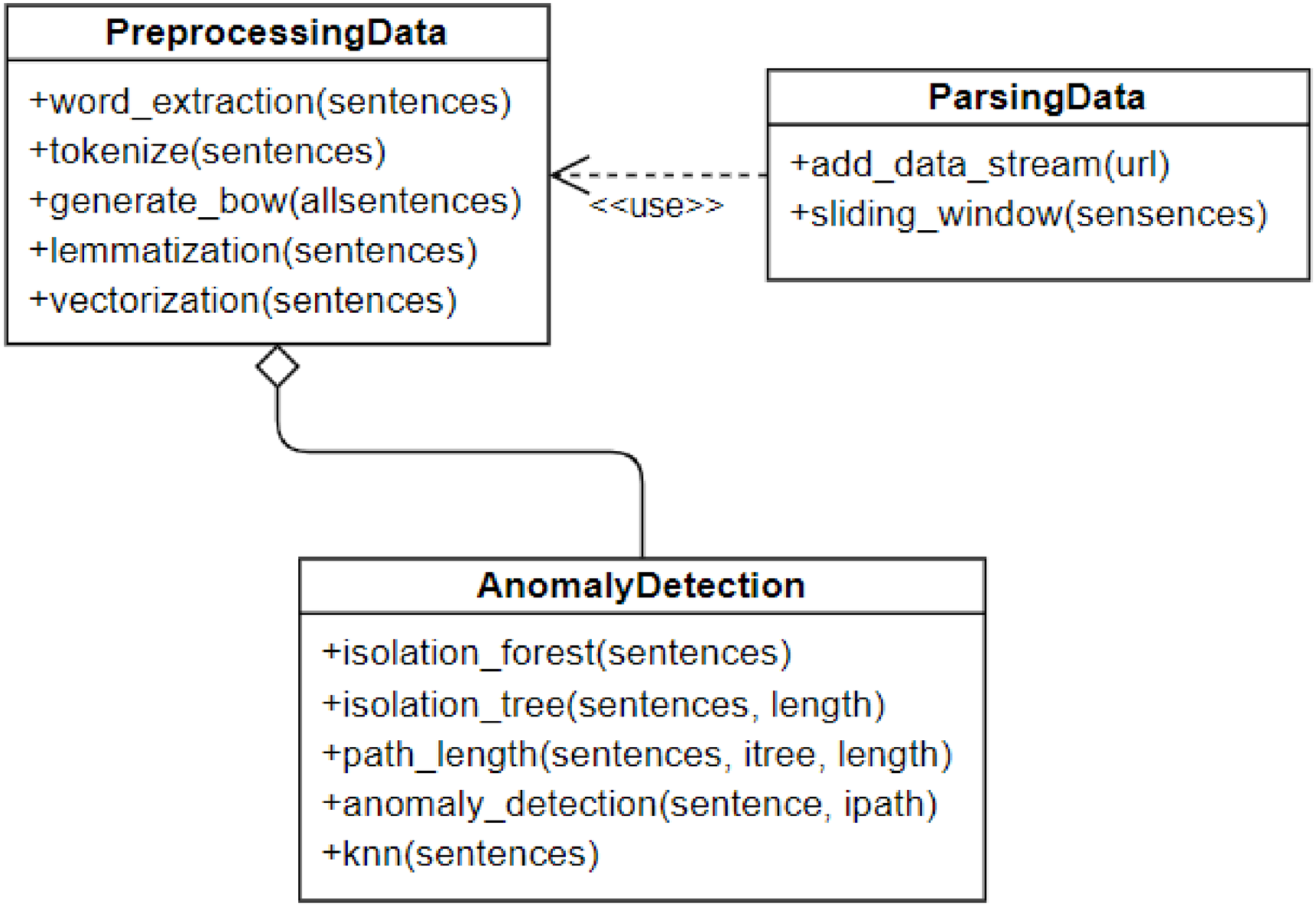
Демонстраційний плакат до магістерської дисертації

Математичне та програмне забезпечення виявлення аномалій в потоках текстових даних

Виконала студентка гр. ІП-з81мп Афанасьєва О.Є.

Керівник Олійник Ю.О.

ДОДАТОК Б СХЕМА СТРУКТУРНА КЛАСІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Демонстраційний плакат до магістерської дисертації

Математичне та програмне забезпечення виявлення аномалій в потоках текстових даних

Виконала студентка гр. ІП-з81мп Афанасьєва О.Є.

Керівник Олійник Ю.О.